

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF ELECTRICAL ENGINEERING  
DEPARTMENT OF CYBERNETICS  
MULTI-ROBOT SYSTEMS



# Radioactive Hotspots Identification by an Unmanned Aerial Vehicle

Bachelor's Thesis

**Kamila Zeleňáková**

Prague, May 2023

Study programme: Open Informatics  
Specialisation: Artificial Intelligence and Computer Science

**Supervisor: Ing. Petr Štibinger**



## Acknowledgments

Most of all, I want to express my gratitude to my supervisor for his helpfulness in every situation and his calmness in solving my problems. I also want to thank my parents for their financial and mental support, my boyfriend whose tireless efforts ensured that I was fully focused on my academic endeavors, my sisters who were always there when I needed them, and last but not least, my friends who supported me even when I didn't believe in myself.

---



## I. Personal and study details

Student's name: **Zele áková Kamila** Personal ID number: **491983**  
Faculty / Institute: **Faculty of Electrical Engineering**  
Department / Institute: **Department of Cybernetics**  
Study program: **Open Informatics**  
Specialisation: **Artificial Intelligence and Computer Science**

## II. Bachelor's thesis details

Bachelor's thesis title in English:

**Radioactive Hotspots Identification by an Unmanned Aerial Vehicle**

Bachelor's thesis title in Czech:

**Identifikace ohnisek radioaktivního záření pomocí autonomního dronu**

Guidelines:

- 1) Familiarize yourself with the Timepix3 radiation event detector and measurement format (<https://github.com/rospix/rospix3>), and the simulation model of the sensor for the MRS UAV system [1].
- 2) Study the available methods for area mapping and data collection by an autonomous robot.
- 3) Design a method for creating a mapping flight path over a user-specified area (convex polygon).
- 4) Propose a data structure to associate the radiation measurements with the corresponding world coordinates based on the odometry of the robot.
- 5) Implement the algorithm [2] to determine the number of radioactive hotspots in the data structure obtained from systematic area mapping.
- 6) Evaluate the proposed method for radiation hotspots identification in a simulated environment and on real-world datasets. Discuss the potential limits of this approach with respect to the number of radiation sources and their mutual distances.

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] Melnykov, V., & Melnykov, I. (2012). Initializing the EM algorithm in Gaussian mixture models with an unknown number of components. *Computational Statistics & Data Analysis*, 56(6), 1381-1395.
- [3] 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.
- [4] Morelande, M. R., & Skvortsov, A. (2009, July). Radiation field estimation using a gaussian mixture. In *2009 12th International Conference on Information Fusion* (pp. 2247-2254). IEEE.
- [5] Newaz, A. A. R., Jeong, S., Lee, H., Ryu, H., & Chong, N. Y. (2016). UAV-based multiple source localization and contour mapping of radiation fields. *Robotics and Autonomous Systems*, 85, 12-25.

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: **27.01.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

\_\_\_\_\_  
Ing. Petr Štibinger  
Supervisor's signature

\_\_\_\_\_  
prof. Ing. Tomáš Svoboda, Ph.D.  
Head of department's signature

\_\_\_\_\_  
prof. Mgr. Petr Páta, Ph.D.  
Dean's signature

### III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her 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

## Declaration

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

Prague, May 26, 2023

.....  
Kamila Zeleňáková

---





## Abstract

The thesis focuses on utilizing an unmanned aerial vehicle (UAV) equipped with a Timepix3 radiation sensor to identify radioactive hotspots. The primary goal of the thesis is to determine the correct number of sources within a predefined search area and accurately determine their locations. For the search itself, the thesis employs an Expectation–Maximization algorithm with a specialized initialization approach based on a Gaussian mixture model. This initialization method estimates the number of components present in the model. For data collection, the UAV autonomously flies through the designated area following a generated flight plan and captures radioactive particles. The position of the drone at the time of particle capture is recorded. Once a sufficient number of particles have been collected, an algorithm is triggered to search for radiation hotspots. The sources can be searched for in real–time during the flight and additionally, after the experiment is completed.

**Keywords** Radioactivity, Unmanned Aerial Vehicles, Timepix3, EM Algorithm, Bayes Information Criterion, Path Planning

---



## Abstrakt

Práce se věnuje identifikaci ohnisek radioaktivity pomocí bezpilotního letounu (UAV) vybaveným radiačním senzorem Timepix3. Práce se zaměřuje především na nalezení správného počtu zdrojů, které se nachází v prohledávané předem definované oblasti a následným určením jejich polohy. Pro samotné vyhledávání se v práci využívá EM algoritmus se speciální inicializací, která pracuje s Gausovským modelem a odhaduje počet komponent, které se v modelu nacházejí. Pro sbírání dat dron autonomně prolétá oblast podle vygenerovaného letového plánu a zachycuje radioaktivní částice. Při zachycení částice se ukládá poloha drona, ve které byla částice zachycena. Když je nasbíráno dostatečné množství částic, spustí se algoritmus na vyhledávání radiačních ohnisek. Zdroje lze vyhledávat v reálném čase i dodatečně po ukončení experimentu.

**Klíčová slova** Radioaktivita, Bepilotní Letouny, Timepix3, EM Algoritmus, Bayesovo Informační Kritérium, Plánování Cesty

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Related works . . . . .	2
<b>2</b>	<b>Preliminaries</b>	<b>3</b>
2.1	Hardware . . . . .	3
2.1.1	Timepix3 . . . . .	3
2.2	Software . . . . .	4
2.2.1	Simulation enviroment . . . . .	4
2.2.2	Robot Operating System . . . . .	4
2.2.3	Rospix . . . . .	5
2.3	Parameter estimation . . . . .	6
2.3.1	Gaussian mixture model . . . . .	6
2.3.2	Expectation-Maximization algorithm . . . . .	6
2.3.3	Bayesian information criterion . . . . .	8
<b>3</b>	<b>Contribution</b>	<b>11</b>
<b>4</b>	<b>Hotspot identification</b>	<b>13</b>
4.1	Mapping path generation . . . . .	13
4.1.1	User input . . . . .	13
4.1.2	Finding intersections of the boundary area . . . . .	13
4.1.3	Generating straight lines . . . . .	14
4.1.4	Order of waypoints . . . . .	14
4.2	Clustering . . . . .	15
4.2.1	Combining radiation data with odometry . . . . .	15
4.2.2	Initial step . . . . .	15
4.2.3	Computing the covariance matrix and mean . . . . .	16
4.2.4	Eliminating fake clusters . . . . .	16
4.2.5	Split detected "islands" of clusters . . . . .	18
4.2.6	An algorithm . . . . .	20
4.2.7	Weighting measurements . . . . .	20
<b>5</b>	<b>Results</b>	<b>23</b>
5.1	Generated data . . . . .	23
5.2	Simulation . . . . .	24
5.3	Experiment . . . . .	25
<b>6</b>	<b>Conclusion</b>	<b>27</b>
<b>7</b>	<b>References</b>	<b>29</b>

---

# Chapter 1

## Introduction

Nowadays, the word robotics is increasingly used in connection with various aspects of everyday life. It provides us with greater convenience and efficiency. Robotic devices can perform tasks faster and more accurately than humans, freeing up our time and allowing us to focus on more meaningful and fulfilling aspects of life. Unmanned aerial vehicles (UAVs), also known as drones, have become one of the most significant technological advances of the 21st century. The term UAV refers to a pilotless aircraft, a flying machine without an onboard human pilot [1]. With increasing accessibility and affordability, they are becoming increasingly popular in everyday life and research activity. UAVs offer significant advantages when used for aerial surveillance, reconnaissance, and inspection in complex and dangerous environments [2].

As mentioned above, UAVs have found applications in situations dangerous to humans. This scenario may include various situations, including potential exposure to ionizing radiation. Several works have already dealt with the problem of identifying radioactive hotspots [3]–[5]. Drones are suitable for this purpose mainly because they can explore a large area in a short time. It reduces survey time and costs while providing comprehensive and accurate radiation distribution data. The ability to detect radioactive materials is provided by various sensors. The Timepix3 [6] hybrid pixel detector readout chip is used in this thesis.

Radioactive decay is often modeled by Poisson distribution, but it can be approximated very well by Gaussian distribution at high mean rates [7]. The Gaussian distribution is a simplification that is well-used in clustering algorithms. Clustering techniques, such as EM [8] or K-means [9], can be used to identify and locate radioactive sources.

When the initial estimation is incorrect, the EM algorithm tends to converge to the local maximum instead of the global maximum. Therefore, it is necessary to find an accurate initial estimate. The EM algorithm alone cannot estimate the number of components in the model. These problems can be efficiently solved using the emEM algorithm [10] in conjunction with the Bayes information criterion (BIC) [11]. The proposed approach involves iteratively executing the EM algorithm with a relaxed termination condition from different initial states. The run that yields the highest likelihood is selected. This process is repeated for different component numbers, and the BIC is computed. The initialization value for EM is determined based on the model with the lowest BIC score.

This thesis introduces an initialization technique for the EM algorithm based on [12] that addresses the challenge of handling Gaussian Mixture Model (GMM) components that are not well separable or partially overlapping. The thesis also includes an algorithm dedicated to generating a flight path for the UAV to map the targeted area systematically. At the end of the thesis, the algorithms are tested on generated, simulated, and real data.

## 1.1 Related works

Unmanned aerial vehicles (UAVs) have become versatile platforms with applications in many domains. Their ability to operate autonomously and collect data from inaccessible or hazardous environments has opened up new possibilities in various fields, ranging from the swarming behavior of UAVs for distributed observation of workspaces [13] to the use of stereo thermal vision for a preliminary inspection of electrical power lines [14]. The exploration of cave environments [15], autonomous flying into buildings in firefighting scenarios [16], multi-UAV systems for target detection [17], and scientific experiments conducted in space [18] are also examined.

The Fukushima disaster in 2011, triggered by a massive earthquake and subsequent tsunami, brought to light the devastating consequences of nuclear accidents. The incident at the Fukushima Daiichi Nuclear Power Plant in Japan resulted in releasing radioactive materials into the environment, posing significant health and safety risks to nearby communities. For this reason, there has been a growing interest in developing technologies to locate and mitigate radioactive sources. UAVs are an effective solution for detecting radiation in challenging areas like forests and paddy fields. With autonomous flight capabilities, UAVs assess decontamination and investigate radioactive material migration in a single flight [19]. In [20], two scenarios are presented with low-cost UAVs equipped with detecting sensors: detecting the radiation level at predefined waypoints and drawing contour maps of radiation levels. Using multiple UAVs at different altitudes enables cooperative detecting and tracking for improved efficiency. In [21], they characterize the radiation field using GMM and investigate the cumulative radiation effects with multiple sources. Other works [22], [23] focus on flying a predefined trajectory close to the ground. The procedures mentioned above do not respond to real-time measured data and thus do not exploit the mobility of UAVs to improve the measurement [24].

# Chapter 2

## Preliminaries

### 2.1 Hardware

#### 2.1.1 Timepix3



Figure 2.1: Picture of MiniPIX TPX3 radiation camera with Timepix3 chip [25].

The Timepix3, developed as an advancement of the Timepix chip [26], offers universal readout capabilities for both semiconductor detectors and gas-filled detectors [27]. The Medipix3 collaboration has financially supported the engineering run of Timepix3. It represents a multi-site project involving prominent institutions such as the European Organization for Nuclear Research (CERN), Nikhef, and the University of Bonn. In this thesis it is used for detection of radioactive particles.

The Timepix3 chip, a 256x256 pixel readout chip organized in 128 double columns, follows a matrix format similar to the Timepix chip. Each double column is further divided into 64 groups of 4x2 pixels, known as superpixels (SP). Each pixel in this configuration, measuring 55 $\mu\text{m}$  by 55 $\mu\text{m}$ , incorporates a fast-response frontend circuit ensuring precise time measurement and high resolution. Each pixel is an autonomous detector unit capable of individually detecting and quantifying particle interactions. Three pixel operating modes can be selected [28]:

1. **Time-of-Arrival (ToA) mode:** It provides precise timing information about the impact time of particles on the detector surface. Specifically, it quantifies the duration between the beginning of the acquisition time and the moment when the particle's signal reaches the predefined Threshold level (THL).

2. **Time-of-Arrival and Time-over-Threshold (ToA&ToT) mode:** It combines the measurement of both the particle's ToA and the deposited energy in a pixel. The measurement of the deposited energy in a pixel from the incident particle is based on the Wilkinson-type Analogue-to-Digital Converter. The sensed energy depends on the number of clock ticks during which the signal is above the preset discrimination level (Threshold). The ToT measurement depends on time, represented by the number of clock ticks required to discharge the accumulated charge in each pixel using a constant current. This mode allows for simultaneous characterization of both time and energy aspects of particle interactions.
3. **Event counting and integral ToT mode:** Events generating a charge above the discrimination level are counted, and their integrated charge is measured within each pixel during the acquisition time.

Timepix3 uses a zero suppression readout technique for data acquisition. It means that only pixels with event data are read out. The chip has two different readout modes [6]:

1. **Data-driven mode:** The chip autonomously transmits data at maximum speed without requiring external commands. Readout is constantly enabled in the SP logic.
2. **Sequential mode:** Data remains stored at the pixel level until an external read command is provided. This command allows parallel reading of columns from 1 to 128.

Timepix3 introduces simultaneous ToA and ToT measurements, extending the capabilities of Timepix. In the data-driven mode, the readout chip sends out a data packet with pixel coordinates, ToA&ToT immediately after the hit is processed by the pixel. During this process, the rest of the chip is still sensitive [29].

The Medipix and Timepix detectors have found extensive applications in X-Ray and Gamma ray applications [30]–[32], in a particle accelerator [33], at International Space Station [34], [35], in miniaturized satellite CubeSats [36] or as such in our case in the field of radiation source localization [4], [5], [37]. For this reason, Timepix3 is used in simulations and real experiments in this thesis.

## 2.2 Software

### 2.2.1 Simulation environment

The MRS UAV System is a full-stack UAV platform with all essential capabilities for research, development, and testing of novel methods. It offers a modular, extensible open-source platform and a complex simulation environment.

Simulation software is a crucial tool for robotic research as it can significantly facilitate the development and testing of algorithms. The mentioned simulation environment makes use of the open-source Gazebo simulator. All UAV hardware elements, actuators, and various sensors are simulated with high fidelity, ensuring coherence with real-world flight and thus accelerating the transition between simulation and reality [38].

### 2.2.2 Robot Operating System

The Robot Operating System (ROS) aims to provide an abstraction over software and hardware components of a robotic system, making applications independent of specific plat-



forms. It acts as middleware between individual components of robotic platforms and user-created applications. This flexibility allows newly developed applications to be easily adapted for robots with compatible hardware features [39].

A crucial feature of ROS is the peer-to-peer transmission of messages between individual software components called nodes. This facilitates the connection of processes across different hosts, ensuring efficient communication and utilizing a name service or master for process lookup. It uses a microkernel design, employing a wide range of small tools for building and running its components, leading to better complexity management and stability. ROS is designed to be language-neutral, supporting multiple programming languages such as C++, Python, Octave, and LISP, allowing developers to choose their preferred language for coding. Additionally, ROS encourages the development of independent libraries for drivers and algorithms, enabling flexibility and avoiding dependencies. The ROS build system effectively performs modular builds within the source code tree and benefits from using CMake, making it comparatively easy to follow this "thin" ideology. Lastly, ROS is a free and open-source framework distributed under the BSD license, permitting the development of both non-commercial and commercial projects [40].

Important aspects of ROS are workspaces, packages, topics, messages, and services. A workspace refers to a set of directories containing a related ROS code set. Packages, including a combination of code, data, and documentation, are located within the workspace. A topic represents a named stream of messages with a defined data type. These topics implement a publish/subscribe communication mechanism. It is important to note that all messages published on the same topic must correspond to the same data type. Services offer an alternative method for nodes to exchange data with each other [41]. A topic is typically utilized when there is a need to transmit data without an expectation of receiving a response, such as in the case of sensor data. On the other hand, a service is employed when a response is anticipated, as seen in situations like path planning.

### 2.2.3 Rospix

Rospix, a minimalistic acquisition software introduced in the article [42], is designed for FITPix and USB Lite interfaces equipped with Medipix2 and Timepix detectors. It allows for parallel communication with multiple interfaces, each measuring asynchronously with different parameters. The output is a single ROS topic with uncalibrated images and acquisition parameter metadata. A set of ROS services controls each interface. Individual parameters such as bias voltage, clock, acquisition time, threshold, and pixel mode are set individually. Rospix offers three modes of operation: single acquisition mode for capturing a single image, batch mode for capturing a predetermined number of images, and continuous mode for capturing images continuously in a loop. ROS nodes running concurrently with Rospix are expected to provide additional functionalities such as visualization, acquisition control, calibration, logging, and track classification.

To ensure communication robustness, Rospix automatically attempts to reset the connection and continue with the acquisition in case of a communication error or interface reconnection. If reconnection fails, Rospix persistently continues its attempts. Additionally, Rospix allows for the resetting of detector configuration before each acquisition to address power supply instabilities that may result in the loss of equalization and corrupted measurements.

The development of the Rospix system was also motivated by the current efforts of the Multi-robot Systems Group at CTU in Prague to perform automated localization and mapping

of radioactive sources using Unmanned aerial vehicle (UAV), where Rospix is being developed specifically for ROS-equipped UAVs to enable onboard data processing and dynamic mission adjustment.

## 2.3 Parameter estimation

### 2.3.1 Gaussian mixture model

The Gaussian Mixture Model (GMM) is a probabilistic model representing a probability distribution as a mixture of multiple Gaussian distributions. GMMs are commonly used for clustering analysis, aiming to assign data points to different clusters based on their similarity or probability of belonging to each cluster. The density function of a GMM describes the probability of observing a data point given the model's parameters. The mean vector  $\mu$  represents the center of the  $i$ -th Gaussian component, indicating the location of the cluster it represents. The covariance matrix  $\Sigma$  characterizes the spread and shape of the distribution.

Definition of multivariate Normal (Gaussian) distribution density function [43]:

$$f(x; \mu, \Sigma) = \frac{e^{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)}}{\sqrt{(2\pi)^k|\Sigma|}}, \quad (2.1)$$

where  $x$  is the point from the dataset,  $\mu$  is the component's mean vector, and  $\Sigma$  is the covariance matrix characterized component.  $\Sigma^{-1}$  represents the inverse of  $\Sigma$  and  $|\Sigma|$  represents the determinant of  $\Sigma$ .

Mixture density

$$g(x) = \sum_{i=1}^K \alpha_i f(x|\mu_i, \Sigma_i), \quad (2.2)$$

where  $\sum_{i=1}^K \alpha_i = 1$ .  $K$  is number of components and parameters  $(\alpha_i, \mu_i, \Sigma_i)$  are the mixture weight, mean vector and covariance matrix of  $i$ -th component.

### 2.3.2 Expectation-Maximization algorithm

GMMs use the Expectation–Maximization (EM) algorithm for parameter estimation. The EM algorithm iteratively estimates the parameters ( $\theta$ ) of a model, starting from some initial condition. Each iteration consists of an expectation step (E–step) and a maximization step (M–step) [44]. In the E–step, it computes the probabilities of data points belonging to each cluster based on the current parameter estimates. In the M–step, it updates the parameters using the weighted data points based on the computed probabilities. Each iteration improves the likelihood or leaves it unchanged. This process continues until convergence, where the parameters stabilize and the likelihood no longer change. In this work, log-likelihood is used to improve the numerical stability. It can be used because the natural logarithm is a monotonically increasing function. The log-likelihood function is defined as:

$$L(\theta) = \log p(X|\theta) = \sum_{i=1}^N \log g(x), \quad (2.3)$$

where  $\theta$  represents the parameters of the GMM,  $X$  is the dataset,  $p(X|\theta)$  is the likelihood function,  $N$  is the number of measurements in dataset  $X$  and  $g(x)$  is the mixture density function of the GMM.

The algorithm described above is shown on figure 2.2 and here:

---

**Algorithm 1:** EM algorithm
 

---

**Data:** Data Set  $X = \{x_1, \dots, x_N\}$

**Result:** Parameters  $\theta = \{\alpha, \mu, \Sigma\}$  and log-likelihood  $L(\theta)$

$\theta, L(\theta) \leftarrow$  initialization;

**while**  $|L(\theta) - L(\theta^{t-1})| > \epsilon$  **do**

    E-step: calculate the new posterior probabilities according to 2.4;

    M-step: calculate the new parameters according to formulas 2.5, 2.6, 2.7;

**end**

return  $\theta, L(\theta)$ ;

---

The EM algorithm is self-consistent [45]. That means the algorithm converges to a stable solution where further iterations do not significantly change the estimated parameters.

Posterior probability

$$\tau_{ij} = \frac{f(x_i|\theta_j^{t-1})\alpha_j}{\sum_{l=1}^M f(x_i|\theta_l^{t-1})\alpha_l}, \quad (2.4)$$

where  $f(x_i|\Theta_j^{t-1})$  is a multivariate Gaussian distribution,  $x_i$  is the  $i$ -th measurement from the dataset,  $\theta_j^{t-1}$  represents parameters of the  $j$ -th Gaussian component at  $(t-1)$ -th iteration,  $\alpha_j$  is the mixture weight parameter of the  $j$ -th Gaussian component and  $M$  is the number of components.

Parameters of GMM [46]:

$$\alpha_j^t = \frac{1}{N} \sum_{i=1}^N \tau_{ij}^{t-1}, \quad (2.5)$$

$$\mu_j^t = \frac{1}{\alpha_j^t N} \sum_{i=1}^N \tau_{ij}^{t-1} x_i, \quad (2.6)$$

$$\Sigma_j^t = \frac{1}{\alpha_j^t N} \sum_{i=1}^N \tau_{ij}^{t-1} (x_i - \mu_j^t)(x_i - \mu_j^t)^T, \quad (2.7)$$

where  $\theta_j = (\alpha_j^t, \mu_j^t, \Sigma_j^t)$  are the mixture weight, the mean vector, and the covariance matrix of the  $j$ -th Gaussian component at the  $t$ -th iteration,  $N$  is the number of measurements in the dataset,  $\tau_{ij}^{t-1}$  is the posterior probability of  $x_i$  from  $j$ -th component at iteration  $t-1$ ,  $x_i$  is the  $i$ -th measurement from the dataset.

The EM algorithm for estimating mixture models offers three attractive features [47]:

1. Both algorithm steps can be computed analytically, eliminating the need for an optimization procedure.
2. The resulting solution ensures that the parameters meet the necessary constraints: the weighting parameters are guaranteed to be positive and sum up to one, while the covariance matrices are guaranteed to be positive-definite.
3. The method is capable of handling missing data. Even if certain elements of a training example ( $x$ ) are missing, the E-step can still utilize the available dimensions to establish

a distribution over the hidden variable. In the M-step, the contribution of this data point is limited to the dimensions of means and covariance matrices where the data were observed.

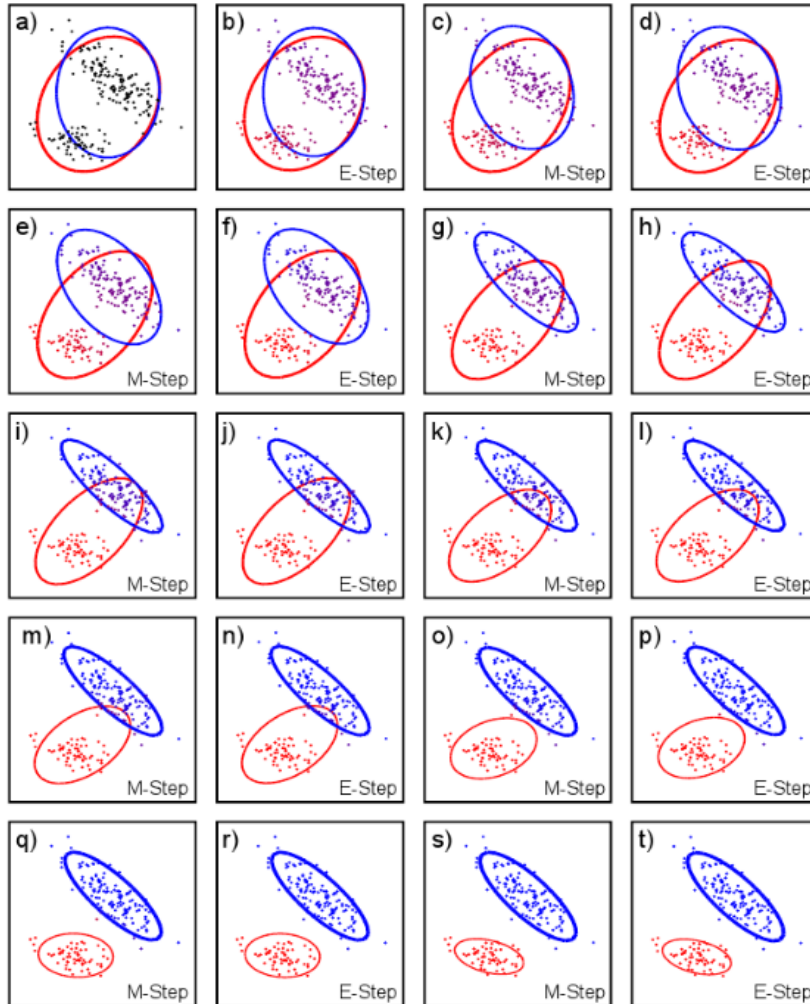


Figure 2.2: Fitting a mixture of two Gaussians to 2D data. a) Initial model. b) E-step. For each data point the posterior probability that was generated from each gaussian is calculated (indicated by color of point) c) M-step. The mean, variance and weight of each Gaussian is updated based on these posterior probabilities. Ellipse shows Mahalanobis distance of two. Weight (thickness) of ellipse indicates weight of Gaussian. d-t) Further E-step and M-step iterations [47]

### 2.3.3 Bayesian information criterion

The Bayesian Information Criterion (BIC), proposed by Schwartz [11], is a statistical criterion employed to select the most suitable model from a collection of candidate models. A candidate is a model that describes data. Each candidate is assumed to be associated with a class of probability distributions with a specific parametric structure [48].

Maximum likelihood estimation tends to favor models with the largest number of components because it maximizes the number of parameters and increases the ability to adapt

to the observed data. However, this approach can lead to overfitting. The BIC attempts to find the model with the highest likelihood while penalizing models with many parameters. Consequently, it favors simpler models in the selection process. It is beneficial in cases where the number of candidate models is large, and the goal is to identify the best model among them.

$$BIC = p \ln(n) - 2 \ln(L), \quad (2.8)$$

where  $p$  is the number of free parameters,  $n$  is the number of measurements in dataset,  $L$  is the log-likelihood function.

The number of free parameters for a GMM:

$$p = (k - 1) + kd + k \frac{d(d - 1)}{2}, \quad (2.9)$$

where  $k$  is number of components,  $d$  is dimension of data and

- $k - 1$  is number of mixture weight,
- $kd$  is number of means,
- $k \frac{d(d-1)}{2}$  is number of parameters in covariance matrix [49].



## Chapter 3

# Contribution

This thesis builds on the research of Melnykov and Melnykov [12] and extends their approach for initializing clustering algorithms, particularly the Expectation-Maximization algorithm. This general algorithm has been adapted for use aboard a drone designed to locate multiple radioactive hotspots. Furthermore, a system for generating a flight plan for the drone was further developed, during which data for this algorithm is collected by the Timepix on-board radiation sensor. The procedure was tested on synthetic data, in a realistic Gazebo simulator, in real experiments with a simulated radiation source.





## Chapter 4

# Hotspot identification

This section presents a detailed solution to the problem of identifying radioactive hotspots by an unmanned aerial vehicle (UAV). The approach is based on [12] but has been enriched with an additional feature. The reason for introducing this feature is the way the data are acquired by the UAV.

### 4.1 Mapping path generation

To locate the source of radioactivity, conducting an aerial survey over a designated area is necessary. During the flight, data collection is performed utilizing the Timepix3 [26] sensor. A comprehensive path-planning strategy is implemented to ensure thorough coverage of the entire area. This strategy focuses on maximizing efficiency by preventing the drone from revisiting any location more than once. The generated path consists of a series of parallel lines, supplemented by connecting segments that link the intersections of these lines with the boundary of the specified area.

#### 4.1.1 User input

The required inputs from the user are:

- The vertices of the mapped area - entered counterclockwise (CCW), only a convex area without holes is considered.
- The mapping density - represents the distance between the parallel lines, entered in meters.
- The mapping direction - represents the direction of line generation, entered in radians, with respect to the parent coordinate frame.

#### 4.1.2 Finding intersections of the boundary area

A boundary area — the line segments connecting the area's vertices — is produced from the specified vertices. All lines and line segments are expressed in parametric form. The first straight line is created from the drone's starting point in the mapping direction. Then, an algorithm for finding its intersections with the boundary follows. The following steps are repeated for each boundary line segment:

1. It is checked to determine whether the current segment is parallel to the generated line.
2. If it is not, use the following equations to express the formula for the parameter to find the intersection

$$xt + X = x's + X', \quad (4.1)$$

$$yt + Y = y's + Y', \quad (4.2)$$

where  $(x, y)$  is the direction vector for the line,  $(X, Y)$  is the point on the line,  $t$  is the parameter by which the direction vector must be multiplied to find the intersection on the line,  $(x', y')$  is the direction vector of the line segment,  $(X', Y')$  is the point where the line segment starts, and  $s$  is the parameter by which the direction vector must be multiplied to find the intersection on the line segment.

Equations 4.1 and 4.2 are parametric forms of the line and the line segment with equal coordinates. The following equation is the resulting expression of the parameter  $s$

$$s = \frac{x(Y' - Y) + y(X - X')}{yx' - xy'}. \quad (4.3)$$

The parameter  $s$  is calculated because it can be used to determine whether the intersection lies on or off the line segment.

3. If the parameter lies in the interval  $[0, 1]$ , the intersection between the boundary line segment and the given line exists.
4. The intersection will be found by multiplying the direction vector by the parameter and shifting the vertex of the line segment by it.
5. If the intersection is duplicated, it will not be considered. Duplication appears when two consecutive boundary line segments have parameters 1 and 0. This case occurs when the intersection is located at the boundary vertex.

The found intersections form a space filling path.

### 4.1.3 Generating straight lines

After slicing the boundary area using the aforementioned algorithm, a new line parallel to the previous line is constructed. The distance between the lines is equal to the specified mapping density. The steps for finding intersections and creating new lines are repeated as long as an intersection is detected. If no intersection is found, new lines are created on the opposite side of the first line, and the algorithm continues.

### 4.1.4 Order of waypoints

Before generating a new line, the program determines whether the newly added intersections are in the correct order. Since the given area must be convex, a line can have at most two intersections. They are sorted according to the last point on the path. The intersection closer to this point will be added to the path first.

The created path is converted to a Path message and sent to the MRS UAV System [38] which constructs a feasible trajectory based on the given path. The examples of a final space filling path are shown in Figure 4.1.

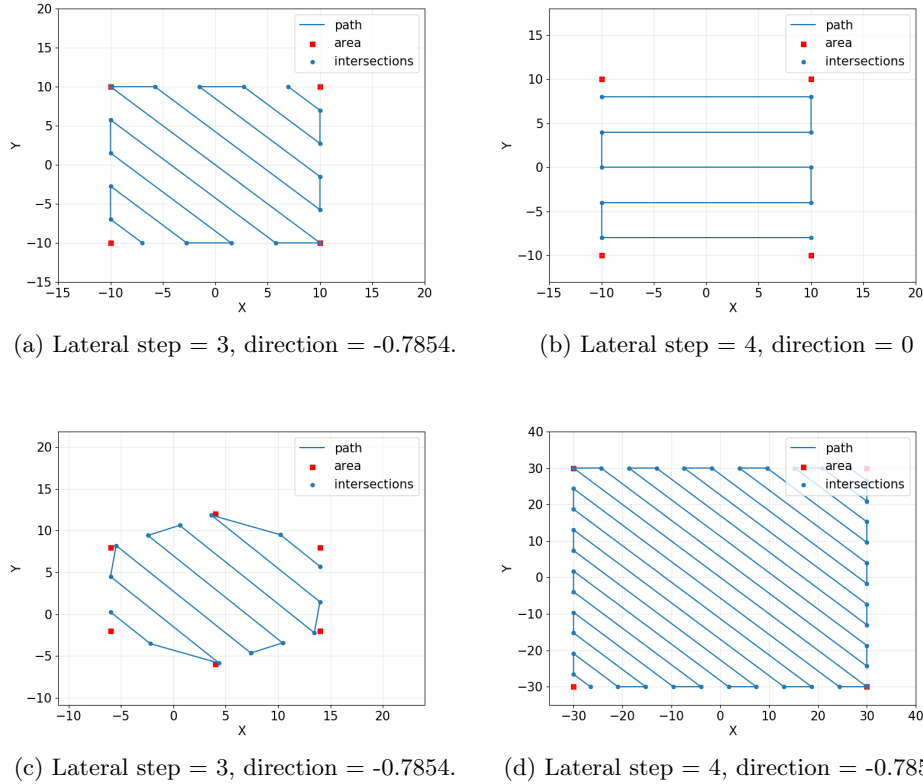


Figure 4.1: Examples of space filling paths created by the developed algorithm from Section 4.1.2. The lateral step represents the distance of parallel lines, and the direction represents the angle in radians. In (b), the direction is  $90^\circ$ , and in (a), (c), and (d), the direction is  $45^\circ$ .

## 4.2 Clustering

The implemented program can compute the covariance matrix using the algorithm from [12] in real time or after the experiment. In addition, it can combine the data from the Timepix3 detector and odometry data. The program only considers a constant mapping height which is specified by the user. Therefore, the data is represented in 2D to reduce the computational complexity. The user can specify the algorithm update rate in Hz.

### 4.2.1 Combining radiation data with odometry

For the proper functioning of the algorithm, it is essential to know where the drone detects the radioactive particle. Therefore, a structure was designed that stores the information about the current particle and the drone's position when the particle was captured. For each particle, its own structure is created.

### 4.2.2 Initial step

In the beginning, a measurement that could be the center of the cluster is selected. For a better result, the initial measurement is not chosen completely at random but according to

the following steps:

1. Calculate the Euclidean distances between all measurements.
2. The measurements are sorted by the sum of the distances to the  $m$  nearest neighbors ( $m$  is a parameter set by the user).
3. The measurements are then ranked from the lowest to the highest sum.
4. The probability of selecting individual measurements is

$$p_i = \frac{\frac{1}{v_i}}{\sum_{j=1}^N \frac{1}{v_j}}, \quad (4.4)$$

where  $p_i$  is the probability of selecting the  $i$ -th measurement,  $N$  is the number of all measurements, and  $v_i$  is the rank of the  $i$ -th measurement.

The selected measurement becomes the center of the initializing circle. Its radius is the distance from its furthest neighbor.

### 4.2.3 Computing the covariance matrix and mean

A rectangle is constructed around the initial circle. Then, the measurements that lie in the rectangle are selected. The algorithm for computing the covariance matrix:

1. Find all measurements that lie in the rectangle.
2. Calculate the mean of all measurements' positions from step 1. The result is a new center of the ellipse.
3. Update the covariance matrix according to the formula:

$$\Sigma = \frac{\sum_{i=1}^N x_i x_i^T}{N}, \quad (4.5)$$

where  $N$  is the number of all measurements from step 1 and  $x_i$  is the position of the  $i$ -th measurement from step 1.

4. Construct the ellipse. Axis radii are scaled according to chi-squared confidence bound (specified by the user).
5. Construct the rectangle around the ellipse.
6. Stop the algorithm if the estimated covariance matrix is the same as in the previous iteration. Otherwise, continue with step 1.

The result of the algorithm is the initial covariance matrix and mean for the new cluster. The mean value sought corresponds to the center of the ellipse. The algorithm steps are shown in Figure 4.2.

### 4.2.4 Eliminating fake clusters

The residual points may create a fake cluster. Hence, it can be assumed that the spurious clusters were identified at the end. The Expectation-maximization algorithm (EM) and Bayesian information criterion (BIC) were used to detect them. Among the various Bayesian motivated methods employed for determining the optimal number of components, BIC is considered the most effective approach for Gaussian Mixture Models (GMMs) [50]. The following algorithm was used to eliminate fake clusters from the model:

$C$  is the set of all clusters.  $c_{last}$  represents the last element in the set of clusters. Two models were created, namely  $C_{full}$  (full model) and  $C_{red}$  (reduced model).  $C_{full}$  includes all

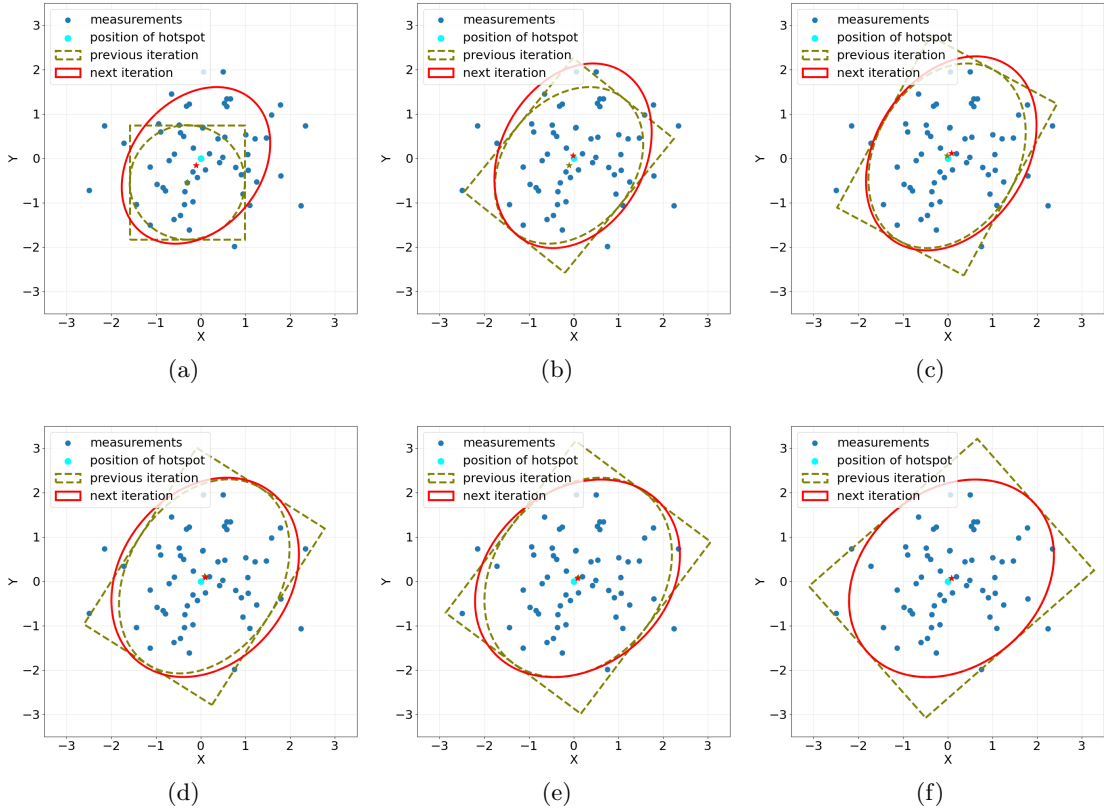


Figure 4.2: The figures show the individual steps of the algorithm. Figure 4.2a shows the first step of the algorithm, which produces an initialization circle with a radius given by the distance of its furthest neighbour. The points that lie in the square drawn around the initialization circle are used to compute the covariance ellipse in the next step of the algorithm. The algorithm repeats until the covariance ellipse changes.

the clusters found in the previous steps of the algorithm.  $C_{red}$  is constructed by excluding the most recently discovered cluster from the  $C_{full}$ .  $bic_{red}$  and  $bic_{full}$  hold the BIC value for both models mentioned above.

The algorithm is executed while the reduced model consists of at least one cluster. Otherwise, it would indicate at most one cluster, and thus no further elimination is needed. During each iteration, the EM algorithm is applied to both models, and their respective BIC values are calculated and compared. The initial value of the mixing parameters is:

$$\alpha_i = \frac{1}{N}, \quad (4.6)$$

where  $\alpha_i$  is the mixing proportion of  $i$ -th cluster, and  $N$  is the number of clusters. Also  $0 \leq \alpha_i \leq 1$  and  $\sum_{i=1}^N \alpha_i = 1$ .

The algorithm stops if  $C_{full}$  has a lower value BIC. Otherwise, the  $C_{red}$  is declared the  $C_{full}$ , and the last cluster of the set is removed from  $C_{red}$ . The result of the algorithm is a set of clusters from which fake clusters have been removed.

**Algorithm 2:** An algorithm for eliminating fake clusters

---

```

Data:  $C = \{c_1, \dots, c_N\}$ 
Result:  $C_{new}$ 
 $bic_{red}, bic_{full};$ 
 $C_{full} \leftarrow C;$ 
 $C_{red} \leftarrow C \setminus c_{last};$ 
while  $size(C_{red}) \geq 1$  do
   $EM(C_{full});$ 
   $EM(C_{red});$ 
   $bic_{full} \leftarrow BIC(C_{full});$ 
   $bic_{red} \leftarrow BIC(C_{red});$ 
  if  $bic_{full} < bic_{red}$  then
    break;
  else
     $C_{full} \leftarrow C_{red};$ 
     $C_{red} \leftarrow C_{red} \setminus c_{last};$ 
  end
end
 $C_{new} \leftarrow C_{full}$ 

```

---

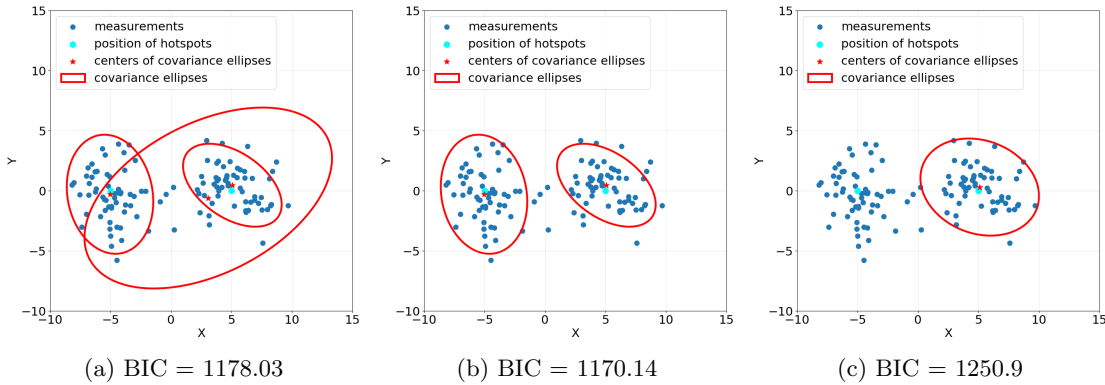


Figure 4.3: Example of elimination of fake cluster. Figures show the different steps of the algorithm. The algorithm reduces the number of components, based on the BIC. When the BIC stops decreasing the algorithm terminates. Subsequently, the model shown in figure 4.3b will be selected in this case.

#### 4.2.5 Split detected "islands" of clusters

In the presence of overlapping or closely situated clusters, the algorithm tends to merge them into a single cluster. Clusters created in this way are called islands. It is necessary to detect and partition such islands into the appropriate number of clusters. Therefore, Algorithm 3 was proposed, which uses a combination of an algorithm called emEM [10] and BIC. In the short run, the EM algorithm terminates as soon as:

$$\frac{L^q - L^{q-1}}{L^q - L^0} \leq 10^{-2}, \quad (4.7)$$

without waiting for convergence.  $L^q$  represents the observed log-likelihood at the  $q$ -th iteration. These short runs are repeated  $x$  times from different randomly chosen points. The run with the best likelihood will be selected, and the complete EM algorithm will be run on it.

---

**Algorithm 3:** An algorithm for splitting “islands”
 

---

```

Data:  $C = \{c_1, \dots, c_N\}$ 
Result:  $C_{new}$ 
for  $i \leftarrow 1$  to  $N$  do
   $C_{act} \leftarrow c_i$ ;
   $C_{best} \leftarrow C_{act}$ ;
   $EM(C_{act})$ ;
   $bic_{act} \leftarrow BIC(C_{act})$ ;
   $bic_{best} \leftarrow bic_{act}$ ;
   $n \leftarrow 1$ ;
  while  $bic_{act} < bic_{best}$  do
     $C_{best} \leftarrow C_{act}$ ;
     $bic_{best} \leftarrow bic_{act}$ ;
     $n \leftarrow n + 1$ ;
     $EM_{short}(C_{act}, n)$ ;
     $EM(C_{act})$ ;
     $bic_{act} \leftarrow BIC(C_{act})$ ;
  end
   $C_{new} \leftarrow C_{new} \cup C_{best}$ ;
end

```

---

The algorithm inspects each cluster for the presence of an island. If an island is found, a split is started. The island is divided into two clusters, and its BIC ( $bic_{act}$ ) is calculated and compared with the best BIC ( $bic_{best}$ ). The preferred model has the lowest BIC. The island is partitioned until the number of clusters with the smallest BIC value is found. Example of algorithm is shown in the figure 4.4.

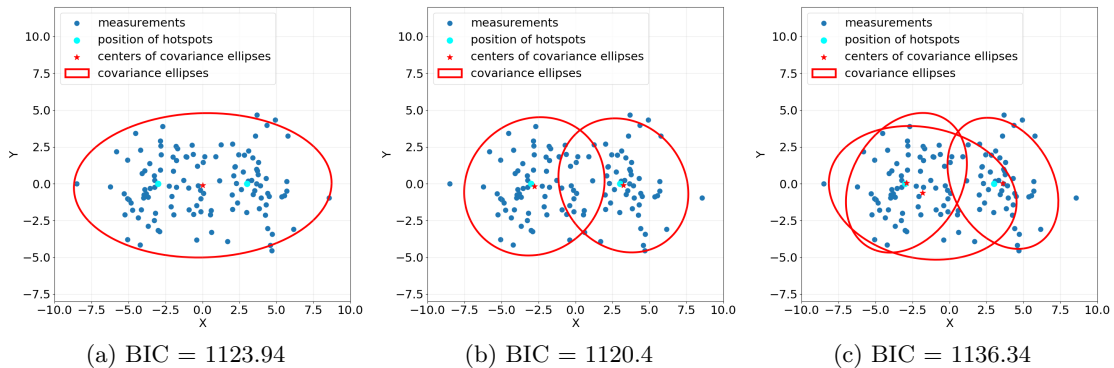


Figure 4.4: Example of splitting. The algorithm shown in the figures splits the component until the BIC of the model stops decreasing. Finally, the model with the lowest BIC will be selected in this case, figure 4.4b.

### 4.2.6 An algorithm

The proposed algorithm utilizes the Gaussian mixture model (GMM) [3] to model the radiation field because it offers a simple and computationally efficient way to describe it. It identifies the quantity and positioning of static radioactive hotspots. The algorithm is designed to work correctly even with sources that have different radiation intensities. The approach can distinguish well-separated sources, where the radiation fields of different sources do not overlap, and in certain conditions also sources whose radiation fields overlap. The algorithm works according to the following steps:

1. Choose the initial measurement that becomes the center of the initial circle and calculate its radius according to subsection 4.2.2.
2. Estimate parameters of the Gaussian mixture model from subsection 4.2.3
3. Exclude measurements that lie within the initial ellipse from the data set. The ellipse is constructed from the parameters calculated above.
4. Go back to step 1. until only a few measurements remain (it is inappropriate to do another cluster).
5. Eliminate fake clusters using the algorithm from subsection 4.2.4.
6. Split the identified cluster "islands" into the appropriate number of clusters according to the algorithm in the subsection 4.2.5.
7. Detect and exclude spurious clusters after splitting.

The last step of the above algorithm searches for the model with the lowest BIC. It compares the BIC of the full model (that includes all found clusters) and the reduced model (that removes only a single cluster while retaining the others). When the BIC of the reduced model is smaller, it is marked as a full model. A reduced model is created as described above and repeats until the model can no longer be reduced. The example of algorithm is shown in the figure 4.5.

### 4.2.7 Weighting measurements

When the drone uses the space filling path from section 4.1, it slows down for each sharp turn. In these sections, measurement density increases regardless of the actual radiation intensity, which leads to distortions in the results. Measurement weighting is introduced to improve the results.

A grid is created with cells across the entire area. Each cell contains the measurements that were captured in this cell sector and the time the drone spent in the sector. The weight (time from the cell that owns the measurement) is added to the structure from section 4.1. A new callback function for `mrs_lib::subscribeHandler`, which receives `nav_msgs::Odometry` messages, is created. It monitors in which cell the drone currently is and for how long it was in which cell. Weighting is used in the calculation of the covariance matrix.

The following formula is now used to calculate the covariance matrix:

$$\Sigma = \frac{\sum_{i=1}^N \frac{x_i x_i^T}{w_i}}{N}, \quad (4.8)$$

where  $N$  is the number of all measurements that lie in the rectangle,  $x_i$  is the position of the  $i$ -th measurement that lies in the rectangle, and  $w_i$  is the weight of  $i$ -th measurement.



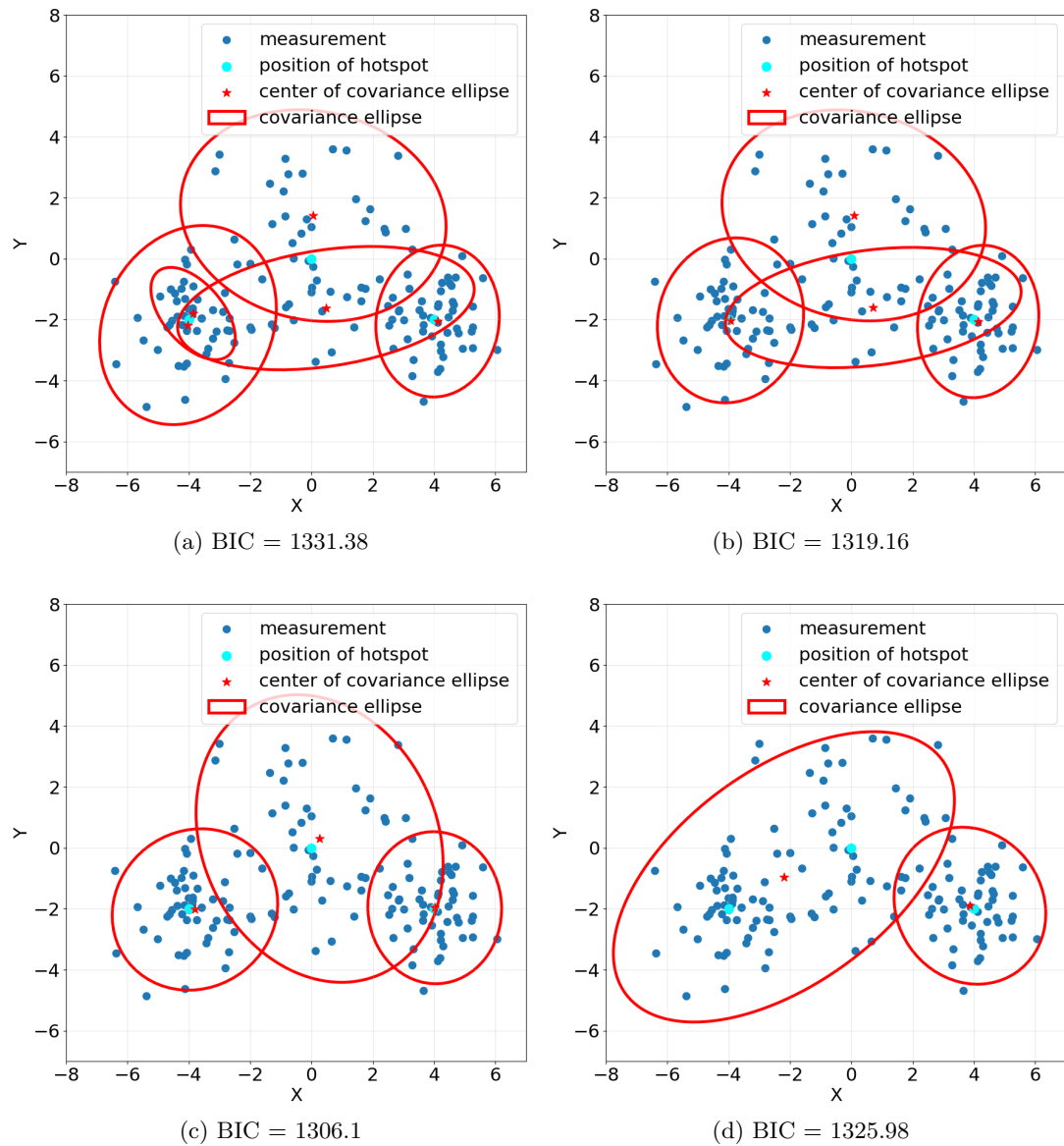


Figure 4.5: Example of elimination after splitting. The algorithm iteratively tries to remove each component and computes the BIC of the reduced model. The resulting model will have the smallest BIC. Figure 4.5a shows the full model and Figure 4.5c shows the model with the lowest BIC, i.e. the best model.



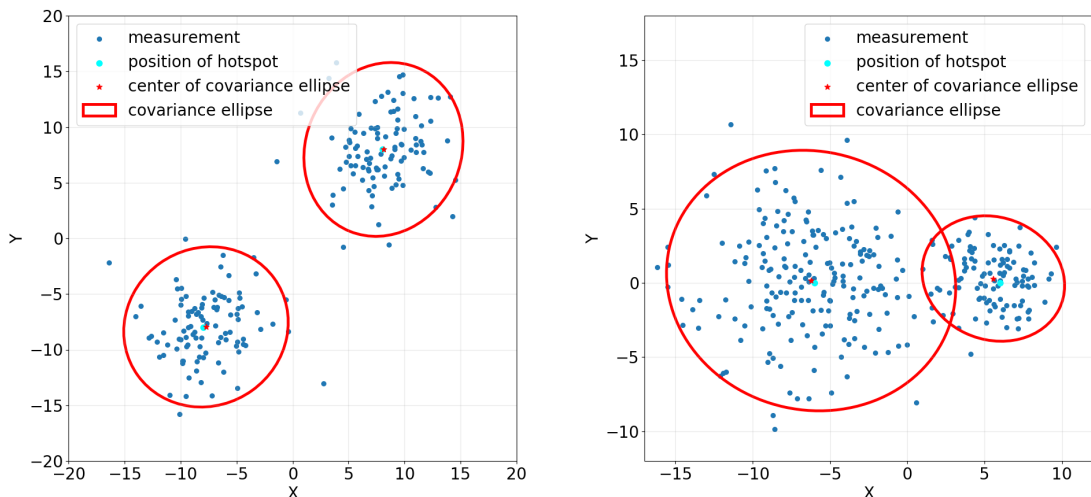
## Chapter 5

# Results

This section demonstrates and tests the functionality of the proposed approach to radioactive source identification. We tested on artificial data, simulated data and datasets obtained from experiments. Americium with various radiation intensity levels were used as radioactivity sources. It was investigated whether the approach proposed in Section 4.2 can identify the number and location of radioactive outbreaks reliably even if the emitters are not well-separated. The approach was tested on artificially generated data, simulation data and real experiment data.

### 5.1 Generated data

A Gaussian Mixture Model (GMM) was used to model the radioactive decay, therefore the initial testing was performed using randomly generated data that fitted the model. The following scenarios were tested: two equally strong sources of radioactivity far apart and overlapping, three equally strong sources far apart, partially overlapping, and finally, two sources of different intensities. Test results are shown in the Figures 5.1 and 5.2.

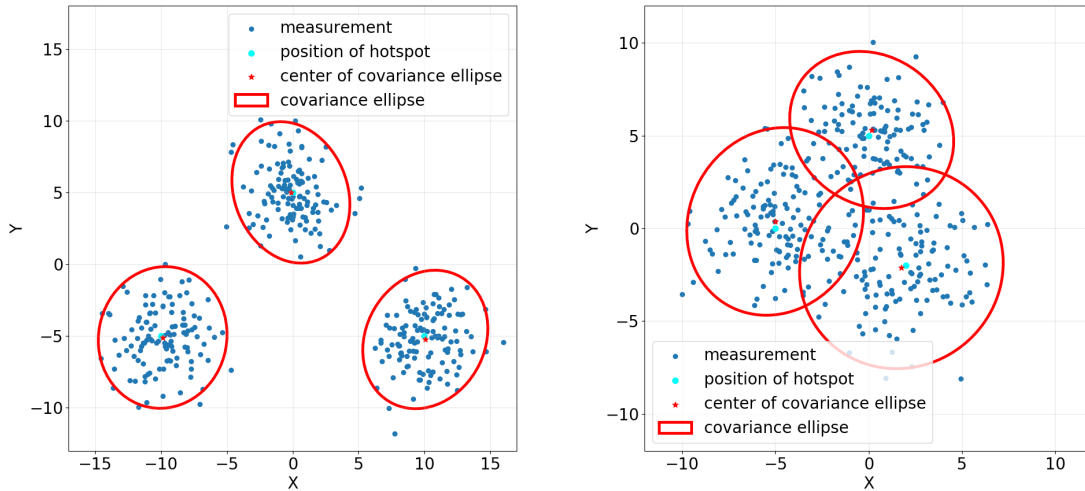


(a) Two well-separated components of the same intensity.

(b) Two partially overlapping components of different intensities.

Figure 5.1: Results of the clustering algorithm presented in Section 4.2 tested on two GMM components.

As is evident from the figures the algorithm worked very well in each case. It managed to estimate the correct number of components and their parameters.



(a) Three well-separated components of the same intensity. (b) Three partially overlapping components of the same intensity.

Figure 5.2: Results of the clustering algorithm presented in Section 4.2 tested on three GMM components.

## 5.2 Simulation

The simulation used the algorithm from Section 4.1 to generate a flight plan for the UAV. Because of this type of path, false clusters were created in the split step 4.2.5. Clusters were created along the lines that the UAV flew through. This was solved by identifying and then penalizing these models in the final selection. Another challenge was that the algorithm created a large cluster at the end that contained all the measurements that did not fall into true clusters. This case is depicted in the Figure 5.3. It was solved by removing the cluster when it was too large. The final results are shown in figures 5.4 and 5.5. The proposed approach was tested on data where clusters partially overlap. Scenarios with emitters of the same and different intensities, were tested.

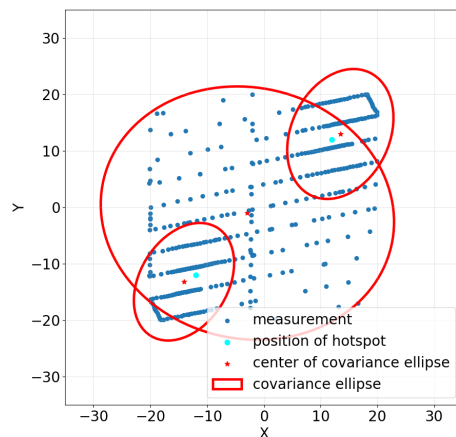


Figure 5.3: Example of misidentification – creating a fake cluster.

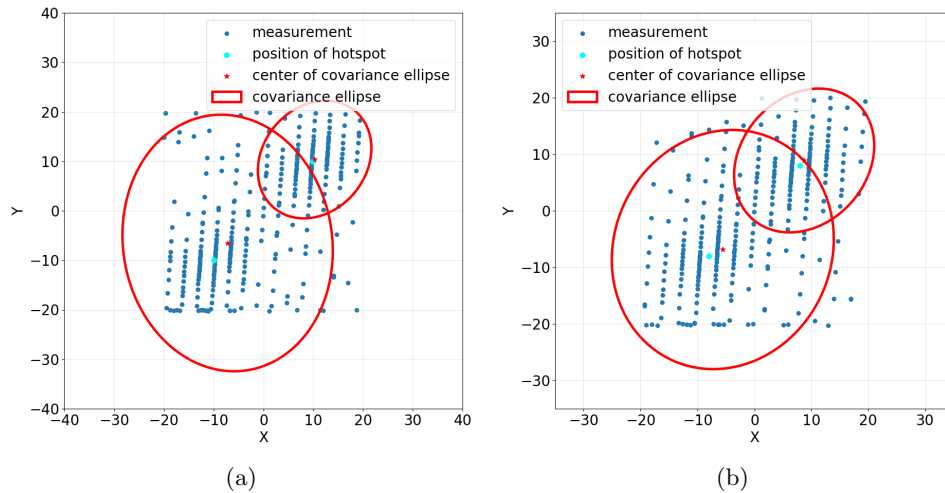


Figure 5.4: Results of the clustering algorithm presented in Section 4.2 tested in simulation. Two partially overlapping clusters with the same intensity.

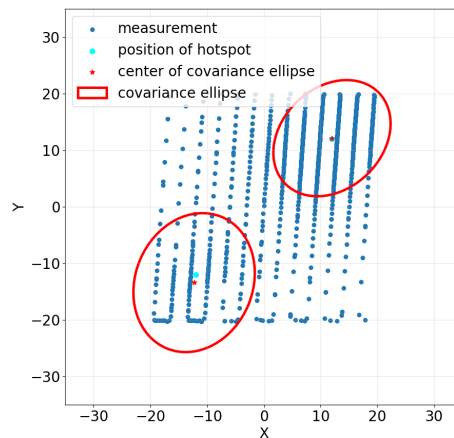
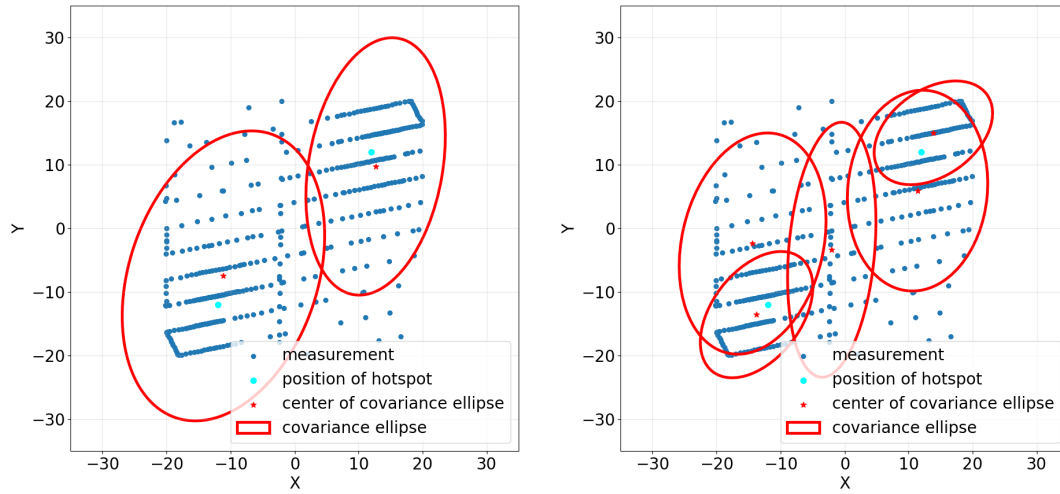


Figure 5.5: Results of the clustering algorithm presented in Section 4.2 tested in simulation. Two partially overlapping clusters with different intensity.

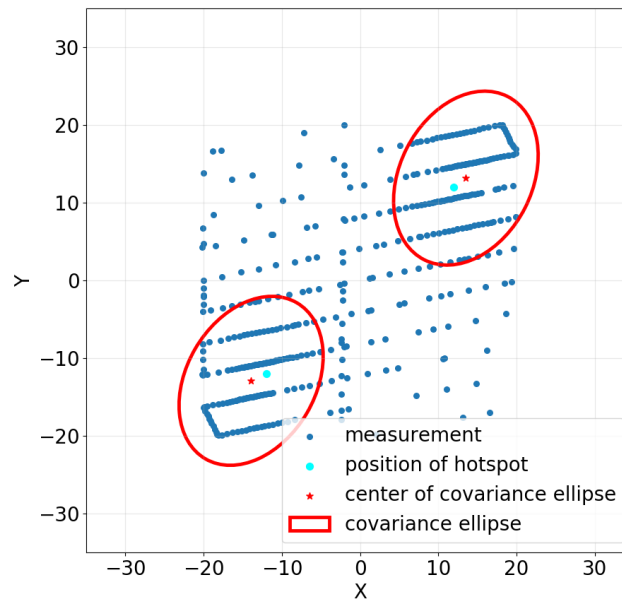
It can be seen from the figures above that the program has identified the number and location of the radioactive hotspots satisfactorily. It had problems with determining the intensity of the sources.

### 5.3 Experiment

The real experiment also used the flight plan generation from the Section 4.1. The experiment used a DJI Flamewheel F450 frame, Intel NUCi7 computer, localization by GPS and Pixhawk4 flight control unit [51]. The results presented in figure 5.6 are performed on the dataset from the experiment showing the steps of the algorithm. The emitters used were of the same intensity.



(a) Result after the initialization step from Section 4.2.2. (b) Result after split step from Section 4.2.5.



(c) Final result.

Figure 5.6: Results of the clustering algorithm presented in Section 4.2 tested on a dataset from a real experiment. Two partially overlapping clusters with the same intensity.

Based on the figures provided, it is evident that the deviation from the actual solution is relatively small.

## Chapter 6

# Conclusion

The aim of the thesis was to search for radioactive hotspots using a UAV equipped with a Timepix3 particle detector. The thesis was based on the research of Melnykov and Melnykov [12] and their clustering algorithm. The algorithm was modified and extended for the needs of UAVs to locate multiple radioactive hotspots. In addition, an algorithm was proposed to generate a flight plan for the UAV for data collection. It enables the drone to cover the entire predefined search area.

Due to the path generation method, new challenges that needed to be addressed:

1. The drone flies at different locations at different speeds. This was solved by weighting the measurements.
2. The splitting step created fake clusters that only covered the area the drone flew (lines). These clusters were identified, and the model that contains them was penalized.
3. In the final step, a fake cluster that contained measurements from both sources was not eliminated. This cluster was identified and removed at the end.

The algorithm was tested on synthetic data in a realistic simulation and a dataset from an experiment with a simulated radioactivity source. The algorithm was verified to perform satisfactorily and can determine the number and location of radioactive hotspots.





# Chapter 7

## References

- [1] K. P. Valavanis, “Advances in unmanned aerial vehicles: state of the art and the road to autonomy,” 2008.
- [2] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. Springer Science & Business Media, 2010.
- [3] M. R. Morelande and A. Skvortsov, “Radiation field estimation using a Gaussian mixture,” in *2009 12th International Conference on Information Fusion*, 2009, pp. 2247–2254.
- [4] P. Štibinger, T. Báča, 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, 2020. DOI: 10.1109/LRA.2020.2978456.
- [5] 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)*, 2021, pp. 338–346. DOI: 10.1109/ICUAS51884.2021.9476766.
- [6] T. Poikela, J. Plosila, T. Westerlund, M. Campbell, M. Gaspari, X. Llopart, V. Gromov, R. Kluit, M. van Beuzekom, F. Zappone, V. Zivkovic, C. Brezina, K. Desch, Y. Fu, and A. Kruth, “Timepix3: a 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout,” *Journal of Instrumentation*, vol. 9, no. 05, p. C05013, May 2014. DOI: 10.1088/1748-0221/9/05/C05013.
- [7] B. Buczyk, *Poisson distribution of radioactive decay*, 2009.
- [8] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.
- [9] J. B. MacQueen, “Some Methods for Classification and Analysis of MultiVariate Observations,” in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1, University of California Press, 1967, pp. 281–297.
- [10] C. Biernacki, G. Celeux, and G. Govaert, “Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models,” *Computational Statistics Data Analysis*, vol. 41, no. 3, pp. 561–575, 2003.
- [11] G. Schwarz, “Estimating the Dimension of a Model,” *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [12] V. Melnykov and I. Melnykov, “Initializing the EM algorithm in Gaussian mixture models with an unknown number of components,” *Elsevier*, vol. 56, pp. 1381–1395, 6 Jun. 2012.
- [13] M. Krizek, J. Horyna, and M. Saska, “Swarming of Unmanned Aerial Vehicles by Sharing Distributed Observations of Workspace,” in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2022. DOI: 10.1109/ICUAS54217.2022.9836073.
- [14] L. Demkiv, M. Ruffo, G. Silano, J. Bednar, and M. Saska, “An Application of Stereo Thermal Vision for Preliminary Inspection of Electrical Power Lines by MAVs,” in *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, IEEE, 2021, pp. 1–8. DOI: 10.1109/AIRPHARO52252.2021.9571025.

- 
- [15] P. Petracek, V. Kratky, M. Petrlik, T. Baca, R. Kratochvil, and M. Saska, "Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021. DOI: 10.1109/LRA.2021.3098304.
- [16] V. Pritzl, P. Stepan, and M. Saska, "Autonomous flying into buildings in a firefighting scenario," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 239–245. DOI: 10.1109/ICRA48506.2021.9560789.
- [17] Y. Stasinchuk, M. Vrba, M. Petrlik, T. Bába, V. Spurný, D. Hert, D. Žaitlík, T. Nascimento, and M. Saska, "A Multi-UAV System for Detection and Elimination of Multiple Targets," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 555–561. DOI: 10.1109/ICRA48506.2021.9562057.
- [18] M. Urban, O. Nentvich, T. Baca, I. Vertat, V. Marsikova, D. Doubravova, V. Daniel, A. Inneman, L. Pina, L. Sieger, R. McEntaffer, T. Schultz, D. Miles, and J. Tutt, "REX: X-ray experiment on the water recovery rocket," *Acta Astronautica*, vol. 184, pp. 1–10, 2021, ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2021.03.019>.
- [19] 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, 2015, ISSN: 0265-931X. DOI: <https://doi.org/10.1016/j.jenvrad.2014.06.027>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0265931X14001994>.
- [20] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost Multi-UAV Technologies for Contour Mapping of Nuclear Radiation Field," *Journal of Intelligent Robotic Systems*, vol. 70, Apr. 2013.
- [21] A. A. Redwan Newaz, S. Jeong, H. Lee, H. Ryu, and N. Y. Chong, "UAV-based multiple source localization and contour mapping of radiation fields," *Robotics and Autonomous Systems*, vol. 85, pp. 12–25, 2016. DOI: <https://doi.org/10.1016/j.robot.2016.08.002>.
- [22] J. MacFarlane, O. Payton, A. Keatley, G. Scott, H. Pullin, R. Crane, M. Smilion, I. Popescu, V. Curlea, and T. Scott, "Lightweight aerial vehicles for monitoring, assessment and mapping of radiation anomalies," *Journal of Environmental Radioactivity*, vol. 136, pp. 127–130, 2014. DOI: <https://doi.org/10.1016/j.jenvrad.2014.05.008>.
- [23] G. Christie, A. Shoemaker, K. Kochersberger, P. Tokekar, L. McLean, and A. Leonessa, "Radiation search operations using scene understanding with autonomous UAV and UGV," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1450–1468, 2017.
- [24] 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)*, 2021, pp. 338–346. DOI: 10.1109/ICUAS51884.2021.9476766.
- [25] Advacam, *MiniPIX TPX3 datasheet*, [online] <https://advacam.com/wp-content/uploads/2022/08/MiniPIX-TPX3-Standard-Datasheet-2022-08-15.pdf>, Accessed: 2022-05-26, 2022.
- [26] X. Llopart, R. Ballabriga, M. Campbell, L. Tlustos, and W. Wong, "Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 581, no. 1, pp. 485–494, 2007. DOI: <https://doi.org/10.1016/j.nima.2007.08.079>.
- [27] V. Gromov, M. van Beuzekom, R. Kluit, F. Zappone, V. Zivkovic, M. Campbell, T. Poikela, X. Llopart, C. Brezina, K. Desch, *et al.*, "Development and applications of the Timepix3 readout chip," *PoS (Vertex 2011)*, vol. 46, pp. 1–8, 2011.
- [28] M. Urban, O. Nentvich, L. Marek, R. Hudec, and L. Sieger, "Timepix3: Temperature Influence on Radiation Energy Measurement with Si Sensor," *Sensors*, vol. 23, no. 4, 2023. DOI: 10.3390/s23042201.
- [29] E. Frojdh, M Campbell, M De Gaspari, S Kulis, X Llopart, T Poikela, and L Tlustos, "Timepix3: first measurements and characterization of a hybrid-pixel detector working in event driven mode," *Journal of Instrumentation*, vol. 10, no. 01, p. C01039, Jan. 2015. DOI: 10.1088/1748-0221/10/01/C01039.

- [30] V Olsansky, C Granja, C Oancea, A Mackova, V Havranek, D Chvatil, and J Bila, “Spectral-sensitive proton radiography of thin samples with the pixel detector Timepix3,” *Journal of Instrumentation*, vol. 17, no. 04, p. C04016, 2022.
- [31] B. Bergmann, T. Billoud, C. Leroy, and S. Pospisil, “Characterization of the radiation field in the ATLAS experiment with timepix detectors,” *IEEE Transactions on Nuclear Science*, vol. 66, no. 7, pp. 1861–1869, 2019.
- [32] C. Fröjdh, D. Krapohl, and G. Thungström, “Hard x-ray imaging and particle detection with TIMEPIX3,” in *Hard X-Ray, Gamma-Ray, and Neutron Detector Physics XVIII*, SPIE, vol. 9968, 2016, pp. 113–118.
- [33] S Lévassieur, B Dehning, S Gibson, H Sandberg, M Sapinski, K Sato, G Schneider, and J Storey, “Development of a rest gas ionisation profile monitor for the CERN Proton Synchrotron based on a Timepix3 pixel detector,” *Journal of instrumentation*, vol. 12, no. 02, p. C02050, 2017.
- [34] D Turecek, L Pinsky, J Jakubek, Z Vykydal, N Stoffle, and S Pospisil, “Small dosimeter based on Timepix device for International Space Station,” *Journal of Instrumentation*, vol. 6, no. 12, p. C12037, 2011.
- [35] M. Kroupa, T. Campbell-Ricketts, S. P. George, A. A. Bahadori, and L. S. Pinsky, “Particle showers detected on ISS in Timepix pixel detectors,” *Life Sciences in Space Research*, 2023.
- [36] T. Baca, M. Jílek, I. Vertat, M. Urban, O. Nentvich, R. Filgas, C. Granja, A. Inneman, and V. Daniel, “Timepix in LEO orbit onboard the VZLUSAT-1 nanosatellite: 1-year of space radiation dosimetry measurements,” *Journal of Instrumentation*, vol. 13, no. 11, p. C11010, 2018.
- [37] T. Baca, M. Jilek, P. Manek, P. Stibinger, V. Linhart, J. Jakubek, and M. Saska, “Timepix Radiation Detector for Autonomous Radiation Localization and Mapping by Micro Unmanned Vehicles,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1129–1136. DOI: 10.1109/IROS40897.2019.8968514.
- [38] 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*, vol. 102, no. 26, pp. 1–28, 1 May 2021.
- [39] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, “The Robot Operating System: Package reuse and community dynamics,” *Journal of Systems and Software*, vol. 151, pp. 226–242, 2019. DOI: <https://doi.org/10.1016/j.jss.2019.02.024>.
- [40] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “ROS: an open-source Robot Operating System,” vol. 3, Jan. 2009, pp. 1–6.
- [41] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System*. O’Reilly Media, Inc., 2015.
- [42] 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, p. C11008, Nov. 2018. DOI: 10.1088/1748-0221/13/11/C11008.
- [43] Y. L. Tong, *The multivariate normal distribution*. Springer Science & Business Media, 2012, pp. 2247–2254.
- [44] R. M. Neal and G. E. Hinton, “A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants,” in *Learning in Graphical Models*, M. I. Jordan, Ed. Dordrecht: Springer Netherlands, 1998, pp. 355–368. DOI: 10.1007/978-94-011-5014-9\_12.
- [45] S. K. Ng, T. Krishnan, and G. J. McLachlan, “The EM Algorithm,” in *Handbook of Computational Statistics: Concepts and Methods*, J. E. Gentle, W. K. Härdle, and Y. Mori, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 139–172.
- [46] Q. Zhao, “Cluster validity in clustering methods,” PhD thesis, 2012, pp. 44–46.
- [47] S. J. Prince, *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

- [48] A. A. Neath and J. E. Cavanaugh, “The Bayesian information criterion: background, derivation, and applications,” *WIREs Computational Statistics*, vol. 4, no. 2, pp. 199–203, 2012.
- [49] U. D. Gupta, V. Menon, and U. Babbar, “Detecting the Number of Clusters during Expectation-Maximization Clustering Using Information Criterion,” in *2010 Second International Conference on Machine Learning and Computing*, 2010, pp. 169–173.
- [50] R. Steele and A. Raftery, “Performance of Bayesian Model Selection Criteria for Gaussian Mixture Models 1,” *Frontiers of Statistical Decision Making and Bayesian Analysis*, pp. 1–29, Jan. 2010.
- [51] D. Hert, T. Baca, P. Petracek, V. Kratky, V. Spurny, M. Petrlik, M. Vrba, D. Zaitlik, P. Stoudek, V. Walter, P. Stepan, J. Horyna, V. Pritzl, G. Silano, D. Bonilla Licea, P. Stibinger, R. Penicka, T. Nascimento, and M. Saska, “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, 2022, pp. 1264–1273. DOI: [10.1109/ICUAS54217.2022.9836083](https://doi.org/10.1109/ICUAS54217.2022.9836083).