Doctoral Dissertation

UAV-based Topographic Mapping and Source Localization of the Radiation Field

Abdullah Al Redwan Newaz

Supervisor: Professor Nak Young Chong

School of Information Science
Japan Advanced Institute of Science and Technology

March 2017

Abstract

In this work, we focus on radiation field that could be generated after the nuclear accident or the attack of Radiological Disperse Devices (RDD) which is commonly known as Dirt bomb. Since harmful radioactive substance unleashed in this situation could cause public health and environmental damage, we want to develop the ability for robots to detect quickly and localize the illicit radiation sources.

There are several ways to get the knowledge of the radiation field. One way is to use the static sensor network distribute throughout the area of interest and iteratively estimate the field. However, static sensor network requires high both deployment density and communication/computational loads to provide good accuracy. The most efficient use of sensor can then be achieved using the aerial robots, where each sensor equipped with the robot can explore many different locations.

This thesis describes work in deploying a single Unmanned Aerial Vehicle (UAV) to map the radiation field, and to localize the radioactive sources. Modern UAVs have gained great popularity in the recent year both in research and commercial platforms. Deploying such UAVs in radiation fields are attractive because they allow using the robotic sensors in unstructured and cluttered environments. Furthermore, for many applications, the mission to be performed is time-limited, meaning that a rapid mapping or localization is required to minimize losses.

We bring together results from our application of four distinct problems in radiation fields. Firstly, we seek the radioactive hotspot in unknown radiation fields, where the robot makes an online path planning with myopic observations. Secondly, we estimate environmental boundaries of unknown radiation fields with an apriori known threshold value. Thirdly, accounting the cumulative effects of the sources, we seek a framework to localize the radioactive sources efficiently. Finally, we attempt to solve another problem of radiation fields that is the determination of multiple regions of interest.

We design path planners that enable the UAV to perform above mentioned tasks. We have implemented a trajectory controller to validate our assumption (related to the robot localization). We also show numerical results of experiments which are demonstrated in simulated environments.

Keywords. UAV, Radiation field, Path planning, Source localization, Contour.

Acknowledgements

First of all, I am greatly indebted to my advisor, Professor Nak Young Chong, for accepting me into his research froup and providing generous support, insightful advice, and kind encouragement throughout my time at JAIST to date. No matter how frustrated I might at times feel when entering his office for progress meeting, I always leave with the renewed energy and determination.

I am most thankful to Professor Sungmoon Jeong for agreeing to be the Second Reader for this thesis and providing the thorough and proper introduction to problems associated with radiation fields. He is always patient with me. Whenever I struggled to grasp some of the theoretical concepts, he kindly explained to me. I am looking forward to our future collaborations together.

I am greatly enjoyed collaborating with Mr. Hosun Lee. He provided immensely useful suggestions on practical implementation of controllers and estimators that will be useful quite beyond the work described herein. I have greatly enjoyed our conversations about a wide range of topics, which often undertook during long hours in the laboratory.

The Robotics Lab has always been a place where people are ready to offer keen criticism and advice on any and all topics, academic or otherwise. For this, I thank Professor Xavier Defago, Professor Francois Pierre Andre Bonnet, Mr. Muhammad Irfan, Mr. Kshitij Tiwari, Mr. Valentin Honore, Ms. Dang Thi Le Quyen, Mr. Nguyen Tan Viet Tuyen, Mr. Pakpoom Patompak.

Finally, my deepest and most sincere appreciation goes to those who are dearest to me: My parents MD. Nazmul Newaz and Sayeda Dilruba Khanom have always shown through both words and actions that their love for their children is unconditional and complete; also my brother Abdullah Al Jannat Newaz, whose hard work and dedication to family are more of an inspiration than he knows. I am grateful to My dearest friend Dr. Shafiul Alam who has always encouraged and supported my efforts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural disasters are sometimes inevitable. Thus, effective emergency relief and disaster recovery coordination can be facilitated by immediate on-the-spot investigations. For instance, if radioactivity levels spike in areas around nuclear power plants, it is very important to find source locations and leaks to quickly characterize the severity of the situation. Similarly, many problems related to environmental monitoring such as the chemical spills or tracking of oil in the sea, exploration of radioactively contaminated area, tracking contaminated cloud or forest fire or harmful algae blooms, monitoring the sea temperature, etc. require localizing the source to accomplish the goal. This goal usually entails efficient mapping, sensing, or searching of environmental phenomena. Applications such as search and rescue seek an effective means to use robotic sensors to monitor a target area. This thesis focuses on the problem of generating plans that enable a single UAV to localize the radiation sources efficiently. The problem of spatial sensing has been advocated extensively in sensor network literature. However, this thesis focuses on source localization while respecting robotics constraints.

Modern Unmanned Aerial Vehicles (UAVs) offer the stable motion performance, with hovering capabilities both in indoor and outdoor environments and also with carrying moderate payloads. This enables them to perform a wide range of application tasks, which include surveillance, search, and rescue, exploration and mapping. The ability to access and navigate in unstructured or cluttered environments makes UAVs attractive platform for disaster recovery tasks.

1.1 Types of Problems

Fig. 1 shows an overview of a static radiation field. There are many problems associated with such fields. However, the main goal of this thesis is to find exploration strategies for UAVs that deals with constrained localization problems. Among the many constrained

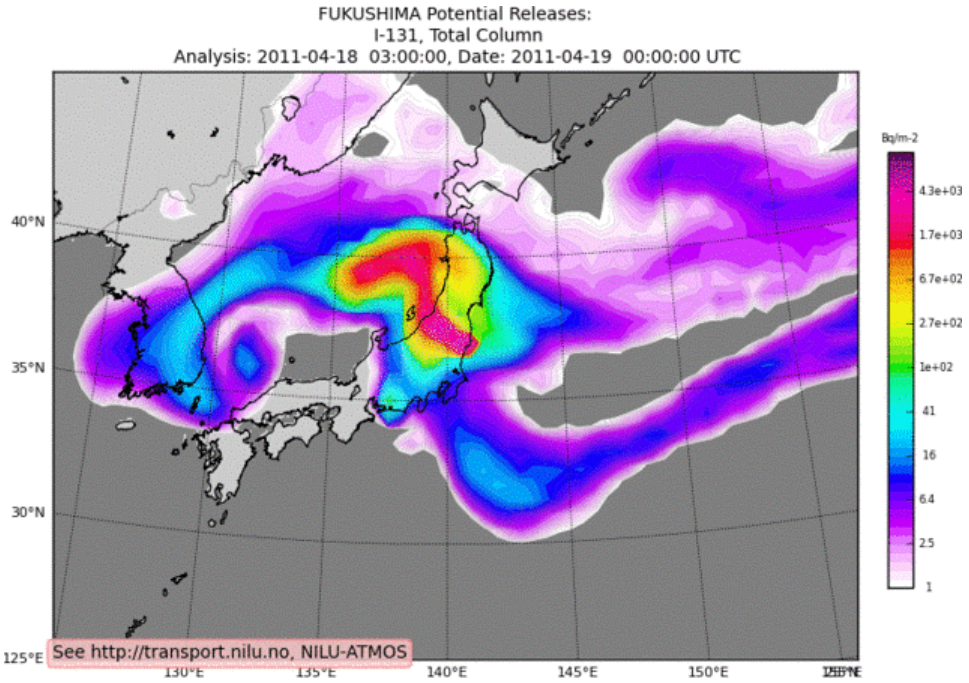


Figure 1.1: An overhead view of the neighborhood around Fukushima nuclear power plant. This picture is an example of a radiation field, and taken from the website of Norwegian Institute of Air Research.

localization problems, we focus on four problem for radiation sources: hotspot localization, environmental boundary estimation, source localization, and multiple regions of interest (ROIs) localization.

1.1.1 Hotspot localization

The radiation hotspot is the zone where the level of radiation is significantly greater than in other regions. The formation of hotspot depends on the geometric dimension and spatial distribution of radiation elements. The localization of hotspot is of great importance to understand the range of radiation effects over contaminated areas. Efficient emergency relief and disaster recovery coordination can be facilitated by immediate hotspot localization using unmanned aerial vehicles (UAVs) that must be accomplished within limited time frame. Since the UAV needs to explore over large areas, an efficient UAV navigation algorithm is crucial for fast hotspot localization.

Hotspot seeking is somewhat similar to the extremum seeking problem which has been widely studied in robotics [1, 2, 3]. The goal is to plan an optimal path in which the UAV can find an unknown hotspot location, while minimizing the exploration cost. Although these approaches are often implemented via gradient-based methods, in practice, it is unlikely to get a significant gradient difference at every exploration step especially in large areas. Likewise, when there is no distinct hotpot, the exploration cannot

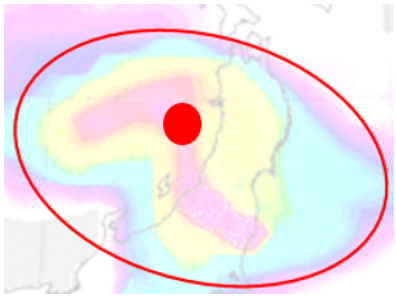


Figure 1.2: An overhead view of the hotspot given the target field which is denoted by red ellipse. The goal of this mission is to seek a path to localize the hotspot (denoted by red dot) while considering the UAV's constraints.

be terminated. To cope with such problems which are caused by similar radiation levels, the Randomly exploring Information Gathering (RIG) algorithm is a worthy candidate [4]. It is computationally efficient and ensures asymptotically optimal solution to achieve information gathering in continuous space with motion constraints. However, when there is a single hotspot, the performance of the RIG should be further improved in such a way that deals with exploration constraints. Specially, the exploration performance can be evaluated using the area coverage metric, which is the amount of sample points required to converge to the hotspot. Furthermore, after localizing the hotspot, a UAV return path should be generated that connects the hotspot and the UAV initial position. This loop closure is necessary to generalize the informative path and the measurement uncertainty assessment of the same locations, to stop exploring due to emergency reasons, etc.

1.1.2 Environmental Boundary Estimation

Since the last decade, estimating the environmental boundary has been drawn much attention in the robotics community. A wide range of applications is possible where robotic sensors have a substantial impact. For instance, robots equipped with dedicated sensors deployed for tracking of oil or chemical spills in the sea [5, 6], localization of radiation sources [7], exploration of radioactively contaminated area [8], tracking contaminated cloud [9] or forest fire [10] or harmful algae blooms [11], monitoring the sea temperature [12], etc. In many such missions are devoted to the gathering of spatial phenomena where sensors observe the measurement in a point-wise fashion at their locations.

Environmental boundary estimation can be thought of as Level Set Estimation (LSE) problems. In this problem, one must find a control policy to identify the region of environment where the measurement of phenomena exceeded some threshold value. Several attempts have been made to accomplish this by utilizing a known map [13,

Figure 1.3: An overhead view of the environmental boundary of the target field. The goal of this mission is to estimate the environmental boundary (denoted by black lines) while tracking the UAV's trajectories.

14]. However, in many practical scenarios such map may not be apriori available. A popular online method is to estimate the boundary of an unknown field with the aid of multiple robots [15, 16]. This approach primarily is benefited from the communication among robots. Since at each time step multiple robots can report the measurement of several locations, computing spatial derivatives of the sensor information performed faster than a single robot. Considering estimation on environmental boundaries instead of the complete area coverage provides a useful abstraction that reduces the energy consumption. Here, the goal is to estimate the shape of the target area subject to the robot's path. However, when the environment is unknown, it is hard to plan such a path that identifies which locations are appealing and which are not.

In this study, we consider the problem of estimating environmental boundary using a single UAV. UAVs offer the stable motion performance, with hovering capabilities both in indoor and outdoor environments and also with carrying moderate payloads. This enables them to perform a wide range of application tasks, which include surveillance, search, and rescue, exploration and mapping. The ability to access and navigate in unstructured or cluttered environments make UAVs attractive platform for a variety of such missions. To achieve this objective, at each time step, the UAV needs sequentially select the sampling locations while respecting to some threshold value. For solving this problem, we propose a boundary estimation algorithm, which utilizes a proportional-integral-derivative (PID) controller to determine the turn rate of UAV [8]. We also provide an optimization technique on the number of samples needed to achieve a certain accuracy by considering to make a closed path. The reason for doing so is that, in the problem such as ours, apart from the gathering of spatial information by traveling to each sensing location, we also have to take into account the time constraints.

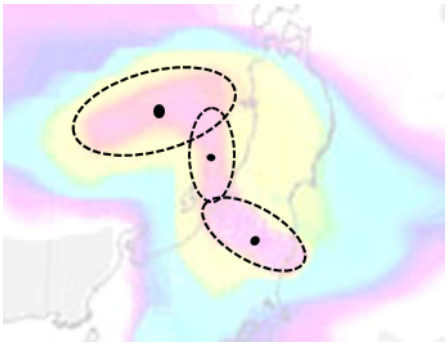


Figure 1.4: An schematic view of estimating radiation sources given the target field. The goal of this mission is to localize radiation sources. The black dots are the radiation sources while the dashed lines represent influenced areas.

1.1.3 Source Localization

After a nuclear accident, a radiation field can be generated by the leakage of radiation sources. A typical radiation field originating from a single hotspot can be generated by three spatial distributions of sources; scattered, clustered and biased. Of these, the clustered sources are relatively easy to localize, because the sources are located in a close proximity to the center of distribution. In other cases, it is not very straightforward, because, when multiple radiating sources generate a hotspot in a cumulative manner, sources do not coincide with the hotspot position. Regardless of our knowledge about the hotspot position, we attempt to solve the multiple radiation localization problem in two steps: the Regions Of Interest (ROIs) selection followed by the source localization. As a disaster recovery plan, it is important to know the distribution of radiation levels over an area of interest, so that rescue mission could be accelerated to minimize the losses. In this kind of situation, autonomous flying robots such as Unmanned Air Vehicle (UAV) can be deployed to monitor the state of radiation effect. Radiation sensors mounted on a UAV can detect the intensity in a radiation field, giving an indication of the activities of nearby sources. The inverse square relationship between the intensity of the radiation source and its distance from the observer can be used to lead the robot to the radiation sources by finding the maximum intensity value.

The search areas may span very large geometric distances, but the measurement attributes of a large radiation field is available only in the close proximity of the sources. Therefore, depending on the radiation leaks, several radioactively contaminated areas can be found in a large radiation field. Since nearby sources also cumulatively contribute to generating a hotspot, without losing the generality, we assume that only a single hotspot exists in each contaminated area, which is caused by all the nearby sources. UAVs may need to fly over a large contamination area, which often leads to problems in designing the exploration strategy with limited resources *e.g.* limited bat-

tery life, sensing range and so on. All locations in a contaminated area are not uniformly important to explore for spatial localization of radioactive sources. Thus, an effective search strategy within a limited fraction of locations can facilitate efficient estimation of radiation source positions. Along this line, it is also important to estimate the distribution of radiation intensity on the geometric map, so that we can reduce our Regions of Interest **(ROIs)** not only for the field characterization but also for the source localization. Given reliable sensor model about the radiation field and accurate localization and navigation performance of the UAV, the goal of this thesis is to plan an exploration strategy for the UAV to rapidly localize all the sources. A common solution is to cover the whole target area so that a global picture of the radiation exposure of that area can be obtained. However, covering the whole area in this regard is not optimal since the UAV has to take in account the limited time constraints (partly due to the battery life).

Recent works estimate the sources in a radiation field using either Hough transformation (HT) [17] or Gaussian mixture [18]. When a radiation field comprises of clustered sources, a standard way is to use the HT, especially when the sources are located at the center of the distribution. HT can significantly reduce the UAV exploration cost, allowing UAVs to determine the source positions by exploring only the contour line of the radiation intensity not far away from the sources [17]. However, the cumulative radiating effect of biased and scattered sources makes the field more complex to estimate, as the sources are not located in close proximity to the center of the distribution. Thus, the problem of estimating a radiation field, which is generated by the combination of multiple sources effect, is often considered as the problem to estimate components from a Gaussian mixture [18].

To balance the tradeoff between the exploration and the localization problems, an adaptive framework is proposed in this work, which can narrow down the robotic exploration and concurrently accelerate the source localization processes. Thus, in this thesis, given the hotspot location along with limited samples of the radiation field, our aim is to answer the following question- *"how quickly and accurately can we localize all the radiating sources in a temporally invariant radiation environment?"*

1.1.4 Regions of Interest (ROIs) Localization

Radiation field monitoring has been commonly studied in robotics. The goal is to plan a path in which the robot can localize all the contaminated locations in a given target area. Since the contaminated locations could be spatially distributed throughout the target area, a search is needed to localize all of them. Thus, required tasks associated with the search inspire various methods in addressing the coverage problem. Techniques used in search can be classified by how many robots are used for this application. In the case

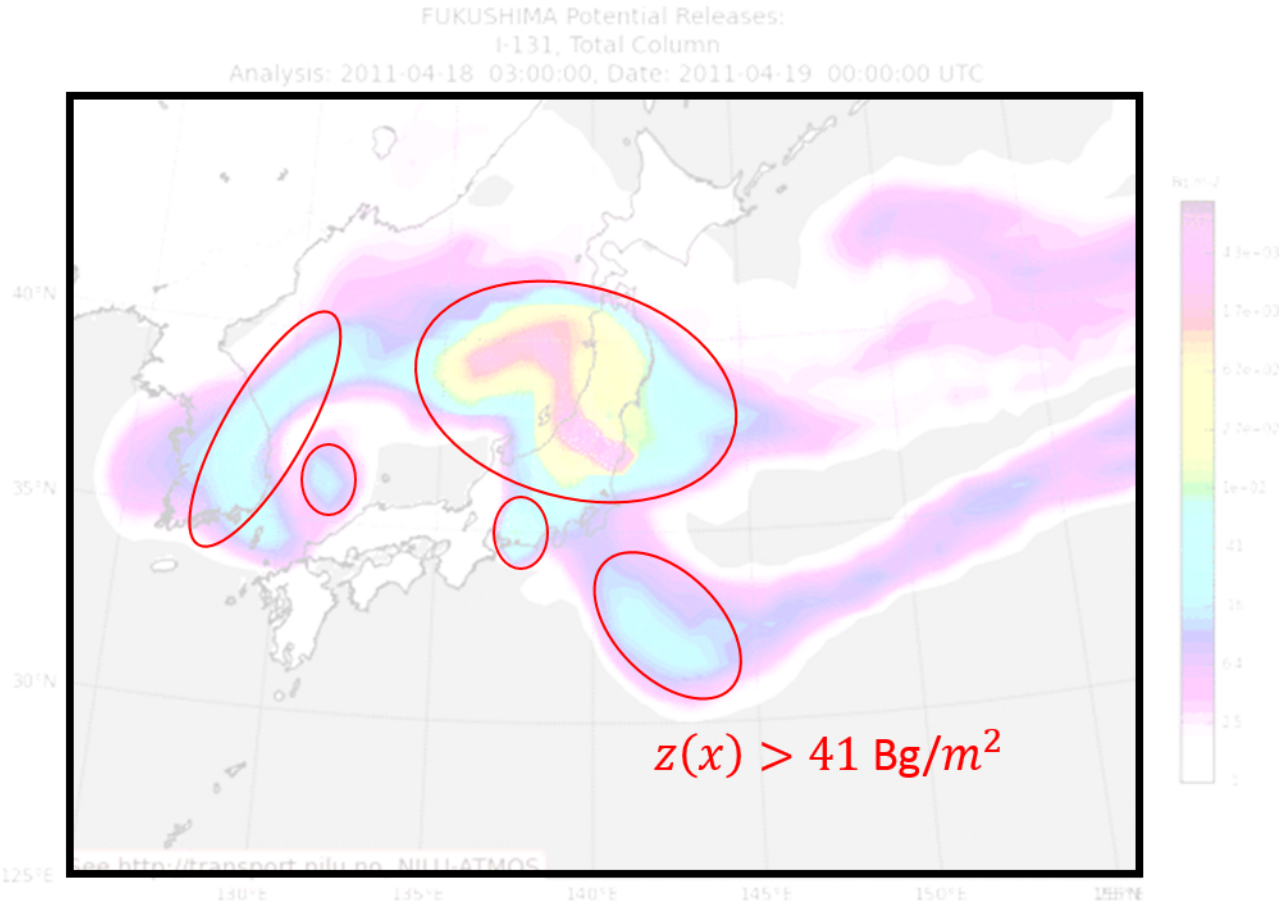


Figure 1.5: An overview of determining Regions of Interest (ROIs) for a large target field. The goal of this mission is to localize ROIs while minimizing the travel cost. The red lines represent ROIs in this figure for a predefined threshold value z.

of multiple robots, the target area can be partitioned in smaller subregions according to the number of robots to reduce the search space. However, in the case of a single robot exploration, neither the partitioning of the target area is straightforward, nor the robot has long sustainability due to the limited battery life.

The majority of coverage planning work has been proposed for known environments [19, 20, 21]. Often these approaches are motivated to minimize the uncertainty metric of a given map. A common choice is to add an exploration to that location where the uncertainty metric such as entropy or mutual information is high. However, in many situations, a radiation map for the target area may not be *a priori* available. The problem can then be seen by its close relation to covering the entire target area for localizing the contaminated locations. Hence, complete coverage algorithms are often used. Even though complete coverage algorithms ensure the complete terrain visitation, lack the opportunity to optimize the localization rate of contaminated locations rather than the coverage objectives.

Considering estimation on environmental boundaries instead of the complete cover-

age provides a useful abstraction that reduces the energy consumption [22, 23]. Here, the path planning problem consists of estimating boundary of contaminated areas that allow the robot to sense the Regions of Interest (ROIs). However, when the environment is unknown, it is hard to plan a path that identifies which areas are interesting and which are not. In conventional algorithms for the coverage planning with obstacles, the path is usually generated to cover the free space of the environment in an optimum fashion. In our problem, rather than avoiding ROIs, we want to identify their locations and geometrical size rapidly. Determining ROIs in a radiation field allows us to prioritize the search area in such way that minimizes the exploration of the robot.

In this study, motivated by a single UAV, we investigate an additional component to the coverage problems by localizing the radiation contaminated locations rapidly. This is important because in a single UAV exploration, sometimes the target area is too large for the UAV to completely cover in a given exploration budget (maximum exploration time). Since it is also of the interest that the UAV is to localize all the contaminated locations as quickly as possible, the algorithm must behave as the complete coverage over long periods of operation. This problem might be thought of as target acquisition problems [24]. However, there is an important caveat. Target acquisition problems assumed that the robot equipped with a sensor that has a wide field of view, whereas in our problem, the robot sensor works in a point-wise fashion. Therefore, the robot needs to travel a location to get a measurement.

1.2 Outline

Our approach focuses on an integrated framework that strives for the complete solution to the radiation field mapping and the source localization. We start with single hotspot based radiation fields. Then, we incorporate an approach to deal with multiple hotspots.

In **chapter 2**, we provide a strong motivation for this research by reviewing some of the efforts done to address each of subproblems.

In **chapter 3**, we explain the preliminaries of this thesis with a prerequisite experiment. This experiment is deemed to verify our assumption.

The following four chapters explain and detail techniques and methods used in our framework.

In **chapter 4**, we address the problem of hotspot seeking in an unknown radiation field where the assumptions is that the field contains only one hotspot.

In **chapter 5**, we develop a framework for environmental boundary tracking and estimation in unknown environments.

In **chapter 6**, we propose an efficient approach to the multiple source localization and contour mapping problem of radiation fields.

In **chapter 7**, we present a framework to solve the problem of determining regions of interest (ROIs) in unknown radiation fields.

Finally, in **chapter 8**, we summarize the thesis contributions, and the capabilities of the proposed framework. We also provide future directions to improve and the extend the current work.

Chapter 2

Related work

The concept of radioactive source localization has been explored extensively in the literature. We present existing work from the areas of the sensor network, active sensing, path planning, motion planning, and environmental monitoring. For each area, we describe how current approaches in the area related to our thesis problem regarding the four dimensions: hotspot seeking, boundary estimation, source localization, and multiple regions of interest determination.

2.1 Hotspot Seeking

In contrast to conventional path planners finding the path between the current position and the goal position, the goal (hotpsot) position is unknown. Since the UAV detects the hotspot only with the intensity measurement, our problem is closely related to the active sampling, selecting observation locations that minimize the prediction uncertainty or maximizing the information gain [25, 26].

Earlier studies were concerned with the localization of sources that do not affect one another [27]. However, the hotspot is no longer coincident with the source position, if the cumulative effect of sources exists. Several strategies attempted to find the hotspot generated by multiple sources [28, 29], which can be generalized into the model-free and model-based approaches. The model-free approaches are extremum seeking methods, where the gradient ascending or the maximum likelihood path is generated. However they tend to converge to local maxima [30]. In the context of model-based approach, source seeking can be performed using either the mutual information (MI) [31, 32] or MI gradient [33]. While such algorithms were shown to be useful for a range of applications, they typically rely on restrictive assumptions on the field and do not explicitly optimize the exploration path.

In the grid based approach, the area is decomposed into a finite number of rectan-

gular cells [34]. A topographic mapping based exploration strategy was proposed to localize multiple sources in our earlier work [35]. Similar approaches can be found in [36]. Probabilistic random sampling based exploration can also be applied when the radiation fields are considered as vector fields [37]. Even though these aforementioned methods perform heuristically well, they overlook the area coverage issues and the exploration was not guaranteed to terminate optimally.

The computational complexity of the planner such as submodular function based near optimal planning [38], maximum entropy sampling [39], maximum mutual information [40] does not suit on-line implementation. Recent path planning algorithms have focused on the generation of approximate paths with limited computation. Our work extends these ideas to the domain of hotspot seeking in a radiation field. Hollinger *et. al.* proposed three variants of a random sampling based information gathering algorithms subject to a budget constraint [4]. Out of their three algorithms, namely, RIG-roadmap, RIG-graph and RIG-tree, they conclude that the RIG-tree is the best in terms of effectiveness.

2.2 Boundary Estimation

Environmental boundary estimation has been recently highlighted in robotics. The goal is to seek a path over a target area in which the robot discovers an isoline in a certain scalar field. The original boundary estimation has been modified and applied in various applications. One way is to deal with this problem is to utilize sensor network over a target area [41]. In this approach, a sensor array deployed in such way that we can get access to the spatial derivatives of the field at every time. However, static sensor networks require high density to provide a good accuracy of observation. Thus, this framework raises two important issues- cost for implementing such an infrastructure, and computational/communication loads for the estimation process.

On the other hand, mobile robots equipped with dedicated sensors can autonomously gather information on the boundary of interest. However, to derive this benefit, a motion planning algorithm is the necessity condition.

Planning algorithm for such systems has gained much popularity in the robotics community. Many methods assume access to a prior map of the target field. The objective of these methods is to utilize the machine learning scheme for level set estimation [13, 14]. In addition to the strong requirement of the availability of prior map, such methods suffer from high computational cost. Hence, when the environment is unknown, the boundary estimation problem becomes more complex.

Several works have been developed to advocate cases in which the boundary is projected by interpolating the robot localization and the sampled points. The methods

used in most studies requires a large number of robots to estimate the boundary shape [22, 42]. The goal in this framework is to design a control strategy for a multi-agent system that has advantages of integrating the detection, tracking and estimation processes. However, in this work, our focus is to solve a different boundary estimation problem by handling a single UAV with a point-wise access to only the value of a general field. In this work, without compromising the estimation accuracy, we intend to generate a closed boundary path that makes efficient use of the limited travel time.

Matveev et al. [12] categorize the solution of such problem into two categories - gradient or derivative dependent approach and other is the gradient-free approach. A gradient-based method is computationally simple and easy to implement especially in the static field. The most popular gradient-based robotic algorithms inspired from the snake algorithm, which often adapted in image segmentation problems [43]. In [15], a decentralized cooperative boundary tracking algorithm proposed that generated an artificial potential field by assuming the direct access to the field gradient. Several works advocated the problem caused by sensor noises to estimate the field gradient, proposed to incorporate a filter into the framework.

The gradient free bang-bang controller reported in [16]. Switching between two pre-defined steering angles proposed in [44]. In recent years, Saldana et al. [45] extended the polynomial approximation [46] to predict environmental boundary behavior for a single robot. Matveev et al. [23] demonstrated a sliding mode method for dynamic fields. Baron et al. proposed PD controller to estimate the contour line in a radial harmonic field [47]. Later, Towler et al. reported PID controller to estimate the contour line in a radiation field accurately [8]. Without knowledge of the boundary evolution dynamics, such methods can efficiently track a dynamic boundary. However, these methods rely, more or less, on the initial approximations/assumptions of the field. In an unknown environment, such assumptions are prone to violate, puts an extra burden on controller tuning, and may a threat of performance degradation.

We propose a novel framework that originates from the control law in [8] and does not employ gradient estimate to track the environmental boundary. We enhance the basic model by incorporating not only a noise canceling filter but also a novel adaptive crossing angle correction scheme. Our method is robust in the sense of minimizing exploration to track the boundary and does not need any prior knowledge on the field. Although we demonstrate our algorithm in a static environment, under the assumption that the environmental dynamics are tractable by vehicle motion, it is then straightforward to implement this method in dynamic environments.

2.3 Source Localization

The radiation field can be analyzed through a wide range of characterization techniques ranging from a point source to mixture models. Earlier works focus on point source based field characterization [48, 27, 49], whereas more variations are found in recent literature. Recent researches have made significant progress in predicting spatial radiation field using Gaussian Process [50, 51]. However, when multiple sources are placed in an area showing cumulative effects, Gaussian Mixture Model **(GMM)** [18] is a well-suited method to characterize it. Of these choices, we use the GMM in our work to characterize a radiation field originating from multiple sources.

In order to illustrate the radiation effects over an area of interest, a radiation map is needed. A grid based map could have a finite number of rectangular cells [52] to represent the field property. On the other hand, we can explain the radiation field using topographic maps [53, 17]. In the topographic map, the field is characterized by large scale intensity measurements and quantitative representation of distribution using contour lines. Thus, it is very useful for the time-limited mission. To this end, our work present the first prediction model of the source locations by analyzing topographic maps.

Hotspot detection is often termed as a source seeking problem in the literature. Several strategies are applied to find a hotspot in an unknown radiation field. Those strategies are mostly divided into two categories, namely, model-free and model-based approaches. Specifically, model-free based approaches involve following a stochastic gradient of the radiation field intensity. It is observed that since the gradient of the intensity is followed, without a priori threshold limit (definition of the hotspot) those algorithms tend to converge to a neighborhood of a local maximum of the field [31]. In the context of model-based approach, source seeking can be performed using either the mutual information (MI) [31, 32] or MI gradient [54, 33]. While the popular approach for the source seeking task is to deploy a group of distributed robots [31, 55, 56], a single robot can travel to several locations in order to gather intensity measurements [57, 58, 59], and then the hotspot can be localized by a predefined threshold value [31]. We exploit the source seeking algorithm to generate a UAV trajectory from an arbitrary intensity zone to the hotspot zone for prior knowledge of the field.

Numerous approaches can be found in the literature to estimate the multiple radiation sources. The Archimedian spiral search pattern [60] is basically exhaustive search to determine the radiation sources within the target area. The Artificial Potential Field (APF) [61] based exploration in a radiation field might get confused more easily with the presence of multiple sources. Multi-robot adaptive sampling uses distributed robot exploration to classify radiation fields via recursive geometric subdivision [56]. If the

map of the target area is a priori known, powerful algorithms like submodular optimization [62], mutual information gain [55], maximum entropy based path planning [63] can yield good results. Despite having a radiation map or an exhaustive search pattern, a key challenge in radiation field mapping is plagued with a limited flight time of the robot needed to find a ROI. This heckles the UAV in large-scale radiation field mapping.

In literature, we have seen that GMM based radiation field parameters can be estimated using a progressive correction technique [18], where a uniformly distributed sensor array was deployed in the area of interest. The sequence of distributions is successively approximated by the Bayes' rule. However, in our case the problem is complicated by the limitation of robotic exploration that gathers spatial measurement attributes of the field. To avoid the problem associated with limited exploration, we propose to use a topographic map to represent the large radiation field with a finite number of contour lines. Only a few efforts have been made to improve radioactive source detection using the topographic mapping strategy. Jerry Towler [17] used Archimedian spiral search patterns to gather measurements and discovered the contour lines with user-defined intensity values. He finally proposed to use the HT to estimate the source position. Although the performance of HT is satisfactory for the clustered sources, in contrast, it gives the worst results for the biased and the scattered sources respectively. Throughour empirical investigation, we demonstrate that our adaptive switching methodology not only optimizes the ROI but also persistently and accurately localizes the sources.

2.4 ROIs determination

Area coverage planning has been extensively studied in robotics. The goal is to seek a path over a target area in which the robot covers all the locations. The basic area coverage has been modified and applied in various applications. The algorithm we present solve a different area coverage problem by generating paths that make efficient use of the limited travel time and maximizing the probability of finding the radiation contaminated locations as the ROIs. This problem is somewhat similar to complete area coverage problems. Coverage planning problem was firstly addressed by Choset [64], where he classified the solution approaches either based on heuristic or cell decomposition.

Heuristic methods explore the target area with predefined rules or a set of behaviors. The widely used heuristic methods are lawnmower pattern, raster scanning, inward spiral search, wall following, etc. Heuristic search is computationally less expensive, but cannot guarantee the optimal performance.

On the other hand, in cell decomposition, the target area is decomposed into smaller areas. Galceran and Carries [65] proposed an exact and uniform decomposition of the

target area by a grid of equally spaced cells. Then, the coverage problem can be solved as the Traveling Salesman problem and is known to be NP-hard. Usually, in that case, a Hamiltonian path is determined using the spanning tree algorithm, which visits each cell exactly once. In recent year, a variant of Hamiltonian path utilized for the persistent coverage problem [66]. However, if there are obstacles in the target area, it is not possible to generate the Hamiltonian path in all the cases. The Boustrophedon cellular decomposition can then solve this problem for bounded planar environments with known obstacles [67]. The key idea is to construct a graph by decomposing the target area subject to obstacle positions and finding a minimal cost tour through all regions. In literature, we have seen an extension of that algorithm while respecting sensor feedback [20, 68, 69]. When unknown obstacles exist in the environment, the Morse decomposition used for determining critical points in the target area, and then incrementally construct the Reeb graph to solve the online coverage problem optimally [70]. Another way is to satisfy a temporal logic specification consisting of safety components in a partially unknown environment [71].

To address the problem of adversarial coverage in environmental monitoring issues, Yehoshua [21] proposed a variant of spanning tree algorithm to split the target area into connected areas of safe and dangerous cells, and then covers safe areas before moving to dangerous one. However, the main flaw of this work is that a map of threats in needed in advance, which may not be available in many situations. On the other hand, if we make an assumption on the distribution of phenomena, for instance, a mine-laying pattern in the minefield is a priori available, the probabilistic demining algorithm [70] could have the option to solve our problem.

In recent years, estimating the shape of environmental boundaries has been a highlighted topic [72, 43, 73, 42]. However, it is not straightforward to couple a boundary estimation algorithm in our problem. The major difficulty is that coverage planning assumes that all the locations have same properties while boundary estimation assumes that a cluster of contaminated locations can be monitored by approximating only a closed boundary. Most of the boundary estimation research consists of two parts 1) defining a threshold to be used as a gradient information, 2) minimizing the square error between the sampling location and the desired threshold to find the robot's trajectory that close the level curve.

Importantly, boundary estimation was not considered in area coverage studies [46, 45]. In coverage planning, the optimality was evaluated only by the coverage volume, not by time. This is because the properties of measurement attributes were excluded from the planning phase. Our approach broadens the optimality definition to take into area coverage together with localization rate of contaminated locations by including boundary estimation in the planning phase.

Recently, some approaches have been evaluated to reduce the search space. Split-merge cells used for a trapezoidal cell decomposition, which proposed for agriculture applications. Partitioning the target area is popular for multi-robot coverage control problems, for instance, Voronoi-based coverage control introduced in [74, 75], the recursive geometric subdivision offered to monitor spatiotemporal field [76]. Of these, the Voronoi-based coverage control is closest to our own in that it considers the size of heterogeneous disk-shaped robots when it partitions the target area. Though it is different from us in that the size of radiation contaminated areas is used instead of heterogeneous robots, which is sampled from a single robot's exploration as well as is computed online.

An important aspect of any path planning is whether the resource costs are optimized or not. Since robots have limited endurance and sensing range, the coverage plan needs to be optimized for finite resources. Aside from the traditional coverage planning where resource costs overlooked, an optimal persistent coverage plan proposed in [77], where authors find a collection of tours for multiple robots that every target is visited by the robots and the minimum frequency of which a target is visited is maximized. In the case of the single robot, a hierarchical planner proposed in [78], where they compute the mode food ratio heuristic to prioritize search regions.

Our work presents the first opportunistic and iterative environmental boundary estimation for area coverage problem. The methodology is evaluated using two different strategies (namely, boundary estimation and coverage planning) within a novel framework that localizes unknown ROIs with an arbitrary initial position of the robot.The novelties of proposed framework in two folds. Firstly, we proposed a novel online framework to integrate environmental boundary estimation and area coverage problem together. Secondly, inspired by existing area coverage approaches, throughout the results, we demonstrate the performances of our two different algorithms namely Voronoi-based coverage and recursive geometric subdivision.

Although proposed framework is applied in the context of field radiation monitoring, our approach is general and can be scaled to other domains where an opportunistic collection of environmental phenomena is necessary.

Chapter 3

Preliminaries

In this chapter we will briefly explain our basic assumptions, mathematical tools which are used in our proposed algorithms, and the prerequisite experiment which is deemed to verify our assumption.

3.1 Assumptions

Throughout the thesis we make the following assumptions

1. **Perfect localization:** We perform the real robot navigation and trajectory tracking experiments in unknown indoor environments, and observed that localization uncertainty are negligible.

2. **Environment representation:** We consider the cumulative effect of nearby radiation sources and represent the environment using mathematical model.

3. **Static radiation field:** We assume that the radiation field behaves like a static field given the duration of mission, and also do not consider the external disturbances for such kind of fields.

4. **Myopic observation:** We assume that the robot senses the environment in pointwise fashion, and there is no an *apriori* map available to the robot. Therefore, the robot needs to explore a location to get the measurement attribute.

5. **Online exploration:** Throughout the thesis we assume that the mission time is sufficient enough for a single UAV exploration, and the computational power is also sufficient enough for the real-time operations.

3.2 Mathematical tools

To develop the algorithms we use the following mathematical tools:

Gaussian Mixture Model (GMM): In point source localization research, most of problems considered to model the environment using the Poisson distribution. Therefore, the hotspot coincidences in the position of source. However, if nearby sources exhibit the cumulative behaviors in the field, then it is not straightforward to model the environment using the Poisson distribution. We then adopt Gaussian Mixture Model (GMM) to characterize such fields. A Gaussian mixture model is a weighted sum of M component Gaussian densities as given by the equation,

$$p(x|\lambda) = \sum_{i=1}^{M} \pi_i \mathcal{N}\left(x|\mu_i, \Sigma_i\right)$$

where x is a $D-$dimensional continuous -valued data vector, π_i are the mixture weights, and $\mathcal{N}$ is a $D-$variate Gaussian distribution with mean vector μ_i and covariance matrix Σ_i. The parameters of GMM collectively represented by the notation, $\lambda = \{\pi_i, \mu_i, \Sigma_i\}$.

Traveling Salesman Problem (TSP): The TSP is often advocated in combinatorial optimization and graph theory. Given the graph $G = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ represents the set of vertices and $\mathbf{E}$ the set of undirected edges, the algorithm finds the lowest-cost tour of G that includes every vertex, $v \in \mathbf{V}$ of the graph just once. This kind of problem is NP-hard problem and there is no known polynomial-time solution. Therefore, approximation algorithms are often used to find the optimal solutions.

Hamiltonian Path: In graph theory, Given the graph $G = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ represents the set of vertices and $\mathbf{E}$ the set of undirected edges, a Hamiltonian path (or traceable path) is a path in an undirected or directed graph that visits each vertex, $v \in \mathbf{V}$ exactly once. The computational complexity of determining the Hamiltonian path problem is NP-complete. Algorithms like 'brute force search', 'dynamic programming', or 'Monte-carlo algorithm' can be used to solve this problem.

Information Gain: Information gain can be thought of as the **Kullback-Leibler** divergence, which is a measure of the difference between two probability distributions. Let $H(X)$ be the Shanon entropy of the random variable X and $H(X|Y)$ be the conditional entropy of random variable X given that the value of Y is known. Then, the information

gain can be defined as follows:

$$I = H(X) - H(X|Y)$$

In general, information gain quantifies the amount of information of X can be obtained through Y.

Voronoi Diagram: Voronoi Diagram is the partitioning method of a plane with n points into a specific subset of the plane such that each subset contains exactly one generating point. In typical Voronoi diagram the set of generating points is *apriori* known. The Voronoi polygons are then constructed such that every point in a given polygon is closer to its generating point than to any other. However, in our case, we randomly initialize the generating points and iteratively update their positions.

Extended Kalman Filter (EKF): The EKF processes the actions and observations of the robot using a joint distribution. It assumes this distribution is Gaussian with the posterior mean μ_t and covariance Σ_t. Given a non-linear process and measurement models, the EKF can be used to linearize them. The overall algorithm can be divided into two folds: prediction step and update step. In the prediction step, given the control signal u_t and the state transition matrix $F_{t|\mu t-1}$, we predict the state and the covariance as follows

$$\mu_t = f(\mu_{t-1}, u_{t-1}),$$

$$\Sigma_t = F_{t-1}\Sigma_{t-1}F_{t-1}^T + Q_{t-1},$$

where Q_t is the process noise. Then the robot make the observation z_t which is subject to measurement noise covariance R_t.

In the update step, first we compute the measurement Jacobian $H_t|\mu_{t-1}$ and kalman gain K_t as follows

$$K_t = \Sigma_t H_t^T (H_t \Sigma_t H_t^T + R_t)^{-1}.$$

After computing the K_t, we update our state and covariance as follows

$$\mu_t = \mu_t + K_t(h(\mu_t, 0) - z_t),$$

$$\Sigma_t = (I - K_t H_t)\Sigma_t.$$

Contour Line: In a topographic map, the contour line represents the isoline along which the measurement has a constant value. Contour lines could be curved, straight or a mixture of both on a map. The configuration of these contours show us the change

in elevation between points. In our case, constructing contour lines in radiation fields can be thought of in two ways. On the one hand, the geometric based approach this problem is considered as curve reconstruction problem: given a finite sample $\mathbf{V}$ of an unknown curve λ, the goal is to construct a graph $G = (\mathbf{V}, \mathbf{E})$ in such a way that two points in V are connected by an edge $e \in \mathbf{E}$ show us the change in elevation between points *iff* the points are adjacent on λ. On the other hand, the gradient vector flow approach the contour line is the separator between two gradient layers.

Variational Bayesian Inference (VB): VB Inference is well known clustering technique that often used in GMM. This is a particular method which aims to find some approximate joint distribution $Q(X;\theta)$ over hidden variables X to approximate the true posterior $p(X)$ subject to the minimum **Kullback-Leibler** divergence $KL[Q(X;\theta)||p(X)]$. In our case, we do not know the number of mixture component apriori. Therefore, it is advantageous to use the VB because it can automatically localize the unknown sources as to determine the number of the mixture components.

3.3 Prerequisite Experiments

In our following chapters, we will describe different path planning algorithms which would be demonstrated in simulated environment. However, we have observed throughout out the implementation that the robot localization is very accurate with negligible uncertainties. In this following section, we will verify our assumption related to the robot localization with the implementation of a trajectory controller for a low cost aerial platform.

3.3.1 System Overview

We propose a system consisting of a commercially available quadrotor equipped with two cameras and a laptop serving as the ground station. In our quadrotor, the front camera has the larger field of view (FOV) compared to the bottom one. Therefore, we use it for the monocular visual SLAM purpose. The quadrotor can stabilize its orientation with the onboard bottom camera. On the ground station, we implement the path planner, the state estimator, and the position controller. In the following, we describe the aerial platform and the software modules.

Aerial Platform

We built our system from selected off-the-shelf components. Our quadrotor is the *Parrot AR.Drone 2.0*, which is a low-cost platform, and with a weight of only $420g$ and a protective hull, safe to be used in public places. This platform is powered by one $1,000mAh$ LiPo battery, which allows a flight time of $30min$. However, we modify neither the hardware itself nor the software running onboard. We communicate with the quadrocopter using wireless LAN. For navigating a large environment, we use *Amped Wireless Range Extender (SR10000)* that extends the range any wireless network by up to $929m^2$.

The onboard computer of *AR.Drone* is equipped with a $3-axis$ *gyroscope* and *accelerometer*, an ultrasound *altimeter*, and two *cameras*. Furthermore, it features an air pressure sensor, a magnetic compass. The front camera is aimed to cover a diagonal field of view of $92\,\mathrm{deg}$ and has a resolution of 640×360, significant radial distortion and a rolling shutter.

Figure 3.1: **Aerial platform:** Our aerial platform is AR.Drone, which is commercially available low cost platform.

The real time image is transferred to a laptop at $30fps$ using lossy compression. On the other hand, the second camera aims downward, covers a diagonal field of view of $64\,\mathrm{deg}$ and has a resolution of 320×240 at $60fps$. Only one of the two video streams can be streamed to the laptop at the same time. The onboard software uses the bottom camera to estimate the horizontal velocity. Since the software uses the optical flow algorithm to estimate the navigational velocity (Nav Vel.), the accuracy of the estimation highly depends on the ground texture and flight altitude.

Software Modules

Fig. (3.2) shows an overview of our system. The software used in our system runs on two different processing units namely the default AR.Drone hardware, and a lap-

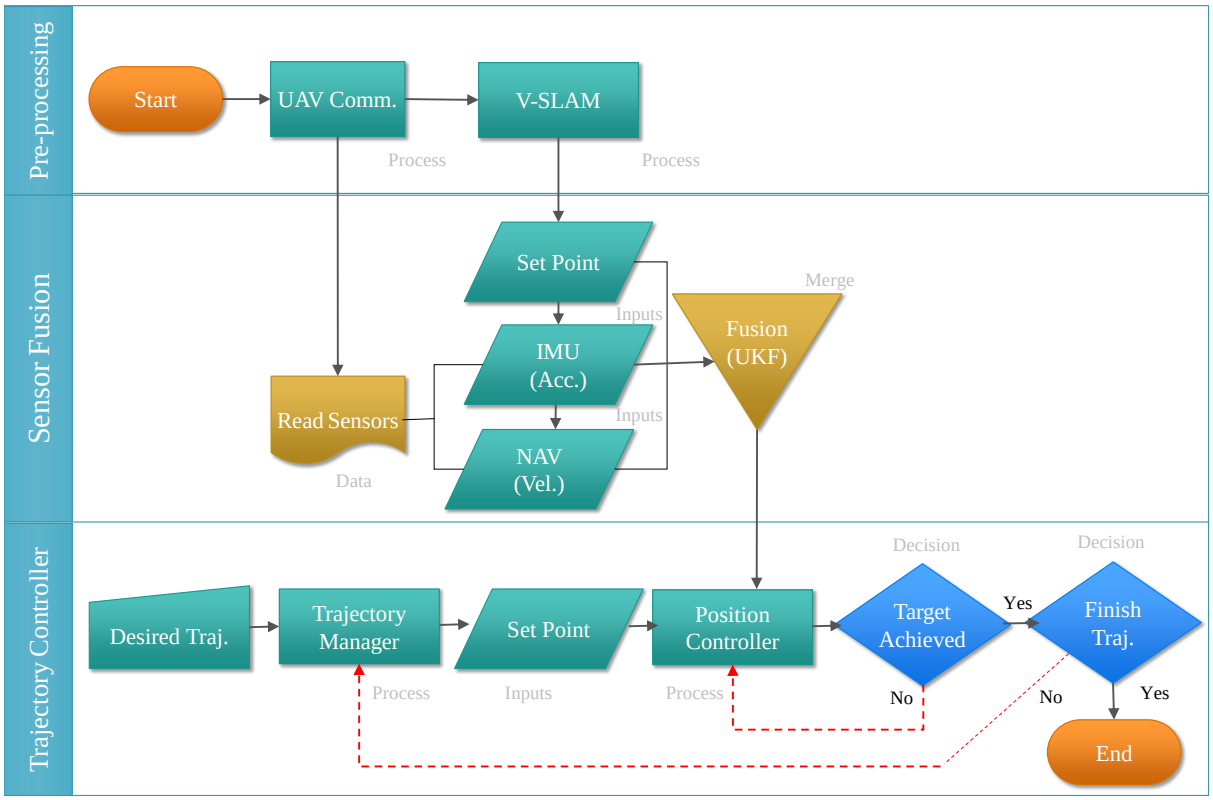


Figure 3.2: Schematic representation of proposed control strategy.

top, which serves as the ground station. All the computations required to localize the quadrotor are performed offboard. We use ROS middleware to communicate with our aerial platform.

All sensor readings, as well as the estimated horizontal velocities, are sent to the laptop at a frequency of up to $200Hz$. Using our setup, the Drone can send image frame to the laptop around 18 fps to 20 fps. On the laptop, we adopt ORB-SLAM as a visual SLAM (V-SLAM) module. The visual odometry pipeline outputs an unscaled pose, which is then fused with the IMU readings in the Unscented Kalman filter (UKF) framework multisensor fusion to compute a metric state estimate.

Given a target area and a set of path, the objective of trajectory manager is to compute the vehicle's motion along a path from start to goal. After deciding the desire trajectory, our actuation phase is responsible to exploit that trajectory. From the state estimate and a reference trajectory, we compute the desired control signals by position controller, which are then sent to the Drone at 30 Hz speed. Note that, the trajectory controller not only control the robot position and corresponding velocity but also control its orientation. It also deals with robot kino-dynamic and algorithmic constraints. Thus, we formulate closed-form solution for our robot navigation problem. One of the fundamental problems in robotics is accurate localization. The noise comes with sensor measurement often leads the state to unstable and poor results. To overcome this

situation, measurements pass through some sort of filters in order to update the belief space. Moreover, if the robot is equipped with multiple sensors to acquire the same state information, sensor fusion can be used to estimate the state with high accuracy [79, 80].

State estimation

While most works perform the state estimation through an EKF [81, 82] in this section, we will show how to implement a UKF to fuse the available sensorial data to track the drone states.

Multiple sensor reading

In order to estimate the six degree-of-freedom (DOF) robot pose, multiple sensor reading *i.e.* both on-board and off-board are fused together. In this work, we have adopted ORB-SLAM algorithm by R. Mur-Artal [83], which is implemented at off-board module. While the SLAM process passively localizes the robot and bulids a map in camera coordinate frame, on-board odometry and IMU actively represent the robot state in body coordinate frame. Note that the output of the SLAM process is *position,* (x_t, y_t, z_t) and *orientation,* $\left(\dot{\theta}, \dot{\phi}, \dot{\psi}\right)$, whereas the output of on-board state is *linear velocity* $(\ddot{x}_t, \ddot{y}_t, \ddot{z}_t)$; *linear acceleration*, $(\ddot{x}_t, \ddot{y}_t, \ddot{z}_t)$ and *angular velocity*, $\left(\dot{\theta}, \dot{\phi}, \dot{\psi}\right)$. It is then important to both of those information into a global coordinate frame.

Since the map is relative to the robot pose, accurate pose estimation is very helpful to understand the environment perfectly. The raw reading from all the sensors can be expressed as follow

$$\begin{aligned} \mathbf{z}^{[1]} &= \{x, y, z, \theta, \phi, \psi\} \\ \mathbf{z}^{[2]} &= \left\{\dot{x}, \dot{y}, \dot{z}, \dot{\theta}, \dot{\phi}, \dot{\psi}\right\} \\ \mathbf{z}^{[3]} &= \left\{\ddot{x}, \ddot{y}, \ddot{z}, \dot{\theta}, \dot{\phi}, \dot{\psi}\right\} \end{aligned} \tag{3.1}$$

where $\mathbf{z}^{[1]}, \mathbf{z}^{[2]}$ and $\mathbf{z}^{[3]}$ are the raw reading from off-board SLAM, on-board odometry and IMU respectively. After determining the appropriate transformation matrices, for instance, from the camera coordinate to the global coordinate T_G^C and from the body coordinate to global coordinate T_G^B, we can easily find the individual raw state such that

$$\begin{aligned} \mathbf{x_{SLAM}} &= T_G^C . z^{[1]} \\ \mathbf{x_{odo}} &= T_G^B . \left\{z^{[2]} \cup z^{[3]}\right\} \end{aligned} \tag{3.2}$$

Since the on-board odometry and IMU both are representing the orientation changes, we can easily summarize the robot pose by taking the mean over those readings. How-

ever,one of the shortcomings of monocular SLAM is that the scale ambiguity. Therefore, to fuse the $\mathbf{x_{SLAM}}$ with $\mathbf{x_{odo}}$, an estimate for this scale factor is essential. This leads to the following representation

$$\mathbf{x_{SLAM}} = \Gamma \cdot \mathbf{x_{odo}}, \tag{3.3}$$

where Γ is scale factor. Once we experimentally determine Γ, at each discrete time frame, k, the raw state measurement is comprised as follow

$$\mathbf{x_k} = \left\{x, y, z, \dot{x}, \dot{y}, \dot{z}, \theta, \phi, \psi, \dot{\theta}, \dot{\phi}, \dot{\psi}\right\} \tag{3.4}$$

In our settings, the measurement accuracy of an off-board SLAM is higher compared to the other sensor modalities. As observer J. Engel [79], the communication speed of video frame rate is much slower than other on-board sensor readings. For this reason, we compute the state transition of the robot pose not only based on SLAM process but also fusing the other sensor measurements that update more frequently. By integrating the actual onboard odometry and IMU measurements, the robot pose is estimated faster and more accurately than a blind prediction on the SLAM reading.

The Unscented Kalman Filter (UKF)

The most popular EKF [84] based fusion algorithm uses recursive mean square error method to estimate the prior and current observation which is represented by Gaussian Random Variable (GRV). The optimal gain of the filter is expressed as through the function of posterior covariance matrices which is then propagated through the first order linearization of a nonlinear system. However, this approximation can introduce large errors in the true GRV which may lead the system to sub-optimal performance and sometimes divergence of the filter. Even though particle filter [85] relax GRV bounds the assumption of the state distribution, it demands more sample points (particles) to reach the accurate distribution. The major flaws of this method are that the number of particles required grows exponentially in the n dimension of the state. Wan *et al.* [86] reported that the UKF results in the approximation that is accurate to the third order for Gaussian inputs for all non-linearity. For non-Gaussian inputs, the approximation is accurate to at least the second order with the accuracy of third and higher order moments determined with the same computational cost as EKF. Therefore, we adapt Unscented Kalman Filter (UKF) for sensor fusion module.

Similar to other version of Kalman filters, the UKF is also a recursive algorithm that is capable to produce very accurate state estimation from the noisy sensor measurements. When the robot executes a control action, $\mathbf{u_k}$, its next state can be estimated by a

Algorithm 1 The Uncented Kalman Filter

Require: weight w_i which size is $2n^a + 1$, sigma points χ_k, initial state x_k^a
Ensure: optimal state x_k with associated covariance p_k

1: Map the nonlinear system
$\chi_{k+1} = \mathbf{F}(\chi_k, k) + B(u_k, k)$
2: Compute the predicted mean
$x_{k+1|k} = \sum_{i=1}^{2n^a+1} w_i \chi_{i,k+1|k}$
3: Compute the predicted covariance
$p_{k+1|k} = \sum_{i=1}^{2n^a+1} w_i \left[\chi_{i,k+1|k} - x_{k+1|k}\right]\left[\chi_{i,k+1|k} - x_{k+1|k}\right]^T$
4: Find the mean observation
$Y_k = h\left(\chi_{k+1|k,}u_k, k\right)$
$y_k = \sum_{i=1}^{2n^a+1} w_i Y_{i,k}$
5: Determine the covariance
$p_y = \sum_{i=1}^{2n^a+1} w_i \left[Y_{i,k} - y_{i,k}\right]\left[Y_{i,k} - y_{i,k}\right]^T$
6: Estimate the cross correlation between state estimation and measurement sequence
$p_{xy} = \sum_{i=1}^{2n^a+1} w_i \left[\chi_{i,k+1|k} - x_{k+1|k}\right]\left[Y_{i,k} - y_{i,k}\right]^T$
7: Determine the Kalman gain
$\mathbf{K}_k = p_{xy}{p_y}^{-1}$
8: update the measurement state
$x_{k|k} = x_{k+1|k} + \mathbf{K}_k(\tilde{y}_k - y_k)$
9: update the measurement covariance
$p_{k|k} = p_{k+1|k} - \mathbf{K}_k p_y \mathbf{K}_k^T$

nonlinear process model such that

$$\begin{aligned} \mathbf{x_{k+1}} &= \mathbf{F}\left(\mathbf{x_k}, w_k\right) + \mathbf{B}\mathbf{u_k} \\ \mathbf{y_{k+1}} &= \mathbf{H}\left(\mathbf{x_{k+1}}, v_k\right) \end{aligned} \tag{3.5}$$

where $\mathbf{F}$ is the state transition model matrix, $\mathbf{B}$ is the control input matrix, $\mathbf{H}$ is the sensor observation model and finally, w_k and v_k represent the Gaussian noise with zero mean estimation. Although there are many complicated prediction models are available for quadrotor UAV in literature, without loss of generality, a working prediction model can be expressed as follow

$$\begin{aligned} \dot{\phi}\left(\mathbf{x}, \mathbf{u}\right) &= K_3\left(K_4 u_{v_x} - \phi\right) \\ \dot{\theta}\left(\mathbf{x}, \mathbf{u}\right) &= K_3\left(K_4 u_{v_y} - \theta\right) \\ \ddot{\psi}\left(\mathbf{x}, \mathbf{u}\right) &= K_5\left(K_6 u_{v_{\dot{\psi}}} - \dot{\psi}\right) \\ \ddot{z}\left(\mathbf{x}, \mathbf{u}\right) &= K_7\left(K_8 u_{v_{\dot{z}}} - \dot{z}\right) \end{aligned} \tag{3.6}$$

Since the acceleration force is propotional to the projection of the quadrotor's z-axis onto the horizontal plane, according to [79], we can rewrite the above Eqn. (5.8) as follow

$$\begin{aligned} \ddot{x}\left(\mathbf{x_k}\right) &= K_1\left(\cos\psi \sin\phi \cos\theta - \sin\psi \sin\theta\right) - K_2 \dot{x}_k \\ \ddot{y}\left(\mathbf{x_k}\right) &= K_1\left(\cos\psi \sin\phi \cos\theta - \sin\psi \sin\theta\right) - K_2 \dot{x}_k \end{aligned} \tag{3.7}$$

where all the proportionality coefficient $K_1, ..., K_8$ are estimated by experimentally.

In order to estimate the optimal state using UKF, we convert the raw state to the augmented state such that

$$\mathbf{x_k^a} = \left[\mathbf{x_k^T}, v_k^T, w_k^T\right] \tag{3.8}$$

The goal of the UKF algorithm is to infer the robot state using a set of weighted sigma points which are deterministically chosen as follow

$$\chi_k = \left[\begin{array}{cc} \mathbf{x_k^a} & \mathbf{x_k^a} \pm \sqrt{\left(L + \lambda\right) p_k^a} \end{array}\right] \tag{3.9}$$

Unlike EKF, for every time step, the sigma points are propagated based on the state transition model and control inputs instead of robot state. After that an optimal state associated with a covariance are computed using the algorithm 1.

3.3.2 Controller Design

We have adapted a nonlinear controller, which is designed with basis on a dynamic model of the AR.Drone, with the closed-loop stability proven using the theory of Lyapunov. In this section, we will explain the simplified mathematical model of our aerial

platform, and then we will show how to use that model to design the control law.

3.3.3 Mathematical Model

In this subsection, we will explain the behavior of our drone using the mathematical model. As our drone is a quadrotor, the flight behavior determined by the speeds of each of the four motors, as they vary in concert, or in opposition to each other. The dynamic model of our quadrotor is already known in the literature, and can be represented as

$$\begin{cases} m\ddot{x} = (\cos\psi \sin\phi + \cos\psi \cos\phi \sin\theta)u_1 \\ m\ddot{y} = (-\cos\psi \sin\phi + \sin\psi \cos\phi \sin\theta)u_1 \\ m\ddot{z} = (\cos\phi \cos\theta)u_1 - mg \\ I_{xx}\ddot{\phi} = u_2 - (I_{zz} - I_{yy})\dot{\theta}\dot{\psi} \\ I_{yy}\ddot{\theta} = u_3 - (I_{xx} - I_{zz})\dot{\phi}\dot{\psi} \\ I_{zz}\ddot{\psi} = u_4 \end{cases} \tag{3.10}$$

where, m represents the mass of the AR.Drone, g is the gravity acceleration, I_{xx}, I_{yy} and I_{zz} are the moments of inertia, and $u_1, \cdots, u_4$ are control signals.

To derive the kinematics model of the drone, we assume the following relationships hold in its body frame

$$\begin{cases} \dot{v}_x = K_1 u_{v_x} - K_2 v_x \\ \dot{v}_y = K_3 u_{v_y} - K_4 v_y \\ \ddot{z} = K_5 u_{\dot{z}} - K_6 \dot{z} \\ \ddot{\psi} = K_7 u_{\dot{\psi}} - K_8 \dot{\psi} \end{cases}, \tag{3.11}$$

where $\dot{v}_x$ and $\dot{v}_y$ represent linear accelerations with respect to the axes x_b and y_b, $\ddot{z}$ represents the linear acceleration with respect to the axis z_w, and $\ddot{\psi}$ represents the angular acceleration with respect to the axis z_w. The parameters $\mathbf{K}_1, \cdots, \mathbf{K}_8$ are proportionality constants to be experimentally identified.

The equation of the motion is then derived by transforming the Eqn. (3.11) to the global frame as follows

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} K_1\cos\psi & -K_3\sin\psi & 0 & 0 \\ K_1\sin\psi & K_3\cos\psi & 0 & 0 \\ 0 & 0 & K_5 & 0 \\ 0 & 0 & 0 & K_7 \end{bmatrix} \begin{bmatrix} u_{v_x} \\ u_{v_y} \\ u\dot{z} \\ u\dot{\psi} \end{bmatrix}. \tag{3.12}$$

To adopt such model, we assume four degrees of freedom of interest (v_x,v_y,$\dot{z}$, $\dot{\psi}$),

and model each of them as an independent linear system. Then we generate the control signals (u_{v_x},u_{v_x},$u_{v_{\dot{z}}}$, $u_{v_{\dot{\psi}}}$) by considering the drone's center of mass as the point of interest to the controller (the target point whose position is being controlled).

3.3.4 Trajectory Controller

We use the nonlinear controller proposed in [87] to guide the AR.Drone in positioning and trajectory tracking tasks. Let the current pose of drone be $\mathbf{X} = [x\ y\ z\ \psi]^T$ and the desired pose be $\mathbf{X_d} = [x_d\ y_d\ z_d\ \psi_d]^T$. We define the tracking error as follows

$$\tilde{\mathbf{X}} = \mathbf{X}_d - \mathbf{X} = \begin{bmatrix} x_d - x & y_d - y & z_d - z & \psi_d - \psi \end{bmatrix}^T. \tag{3.13}$$

Note that $\mathbf{X_d}$ can be a function of time, we can then characterize the trajectory as follows

$$\ddot{\mathbf{X}} = \mathbf{f}_1\mathbf{U} - \mathbf{f}_2\dot{\mathbf{X}}, \tag{3.14}$$

where

$$\ddot{\mathbf{X}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\psi} \end{bmatrix},\ \mathbf{U} = \begin{bmatrix} u_{v_x} \\ u_{v_y} \\ u_{v_{\dot{z}}} \\ u_{v_{\dot{\psi}}} \end{bmatrix},\ \dot{\mathbf{X}} = \begin{bmatrix} v_x \\ v_y \\ \dot{z} \\ \dot{\psi} \end{bmatrix}, \tag{3.15}$$

with $f1$ and $f2$ being the two 4×4 matrices of (3.12). To compute the control law, we adopt an inverse dynamic controller given by

$$\mathbf{U} = \mathbf{f}_1^{-1}(\nu + \mathbf{f}_2\dot{\mathbf{X}}), \tag{3.16}$$

with

$$\nu = \ddot{\mathbf{X}}_d + \kappa_p\tilde{\mathbf{X}} + \kappa_d\dot{\tilde{\mathbf{X}}}, \tag{3.17}$$

where K_{p_x} and K_{d_x} are matrices that represent the proportional and the derivative gains as follows

$$\kappa_p = \begin{bmatrix} K_{p_x} & 0 & 0 & 0 \\ 0 & K_{p_y} & 0 & 0 \\ 0 & 0 & K_{p_z} & 0 \\ 0 & 0 & 0 & K_{p_\psi} \end{bmatrix}, \tag{3.18}$$

and

$$\kappa_d = \begin{bmatrix} K_{d_x} & 0 & 0 & 0 \\ 0 & K_{d_y} & 0 & 0 \\ 0 & 0 & K_{d_z} & 0 \\ 0 & 0 & 0 & K_{d_\psi} \end{bmatrix}. \tag{3.19}$$

Note that it is beyond the scope of this subsection to prove the stability of above controller, our overall approach is to exploit the control law to directly control the quadrotor. The stability of this controller has already been proven in [87].

Experiment Results

We perform a series of real world experiments to validate the properties of the proposed system. All the experiments were performed in indoor environments. In this section, first, we will analyze the V-SLAM algorithm's accuracy and behavior. To obtain the ground truth scale, we performed several tests by manually moving the drone a fixed distance and comparing the moved distance with the distance measured by the V-SLAM system. We also point out the fact that for controlling a drone, it is necessary to have enough visual features in the environment to accurate estimate the pose of the drone by V-SLAM. We remark that the tests in the following section. Finally, we explain how the addition of accurate state estimator significantly improves the control accuracy and tracking abilities.

Scale estimation

It is important to find the accurate scale factor, Γ in Eqn. (3.3) before fusing all the sensor information. On the one hand, to determine the XY scale of the visual SLAM, we performed a manual experiment, where the robot is translated around $0.5m \times 2m$ rectangular area in a hand-held manner, shown in Fig. 3.3. The ORB-SLAM module has the capability to initialize the visual SLAM based on initial motion automatically. When the environment is planar, nearly planar or there is low parallax, ORB-SLAM module explained it by a homography matrix. Otherwise, a fundamental matrix is computed to explain non-planer environment. In our experiments, the non-planar environment is larger than the planar environment. Observing how the scale can be varied in two different environments from a series of experiments, we have found two values for Γ as follows

$$\Gamma = \begin{cases} 5 & \textbf{if planar} \\ 15 & \text{otherwise} \end{cases}. \tag{3.20}$$

On the other hand, we determine the altitude of the drone by fusing the height z

(a) Initial view (b) Final view (c) 3D map

Figure 3.3: Experiment of determining the tracking threshold Value.

from the V-SLAM module, acceleration reading from the IMU, and altitude reading from the ultrasonic sensor. Since the V-SLAM initial height with accurate Γ also does not the coincidence to global origin $(0, 0, 0)$, we initially compute an offset of visual SLAM w.r.t ultrasonic reading. We add this offset to the estimated height which is computed by V-SLAM module, and then fuse all the information using the UKF. All the modules in

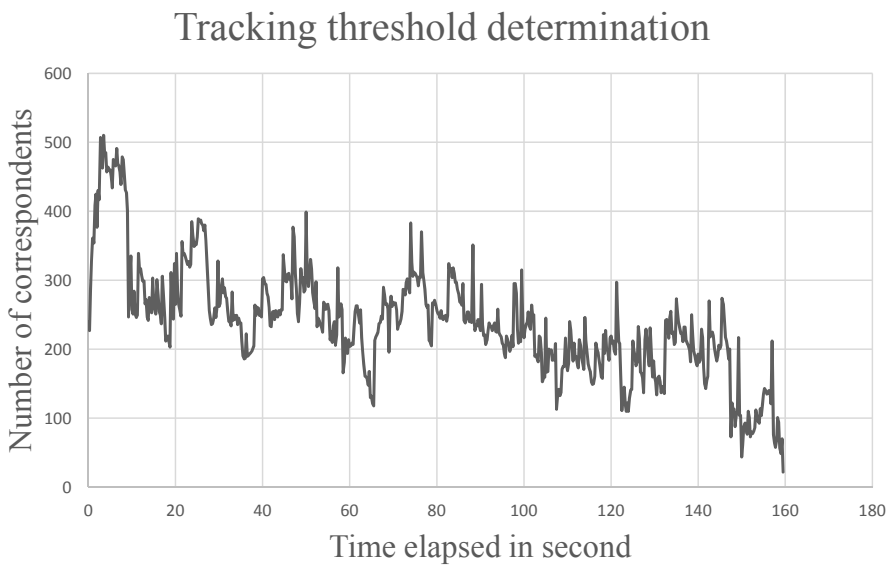


Figure 3.4: Schematic representation of proposed control loop.

Fig. 3.3 should work simultaneously to obtain the best performance from the state estimator. However, we have observed that when the ORB-SLAM module significantly gets reduced the number of feature correspondents, it leads the V-SLAM module to deadlock situation. This fact is known as the tracking lost and a relocalization is possible by backtracking the previous trajectory. To avoid such problem, we initially find the minimum number of tracking point is required to conduct an experiment smoothly. We performed an experiment where the robot was navigating manually in hand hold fashion. We plot the number of feature correspondence with respect to elapsed time

Table 3.1: Overview of experiments

Parameter	**Value**
# waypoints	20
path len.(m)	10.35
max RMSE(m)	0.2308
max var. (m)	0.0120
time (s)	104.1298

in Fig. (3.4), and determine the threshold value just before the tracking lost. In our experiments, we use the threshold value 80.

Trajectory control accuracy

We performed real-robot-experiments in the context of determining uncertainties for the trajectory control. The trajectories generated by the simulated environment were used for this experiments. To this end, we want to minimize the localization uncertainties of the robot given a planned trajectory. Localization uncertainties are relevant since the sampled locations are often inaccurate due to navigational errors from the position controller. We use the root mean squared error (RMSE) to measure the localization uncertainties. While executing a planned trajectory, the RMSEs were computed between the desired positions and the actual positions of the UAV.

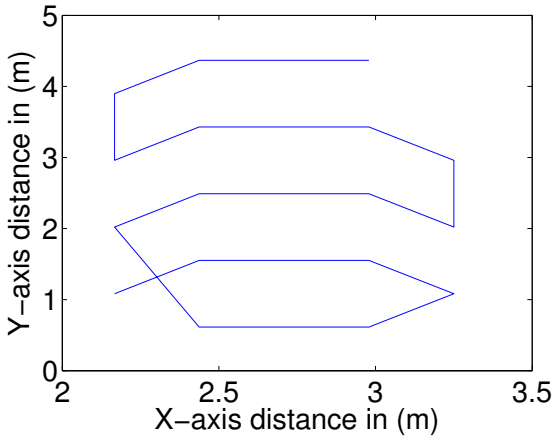
$$\text{RMSE}_i = \sqrt{[\mathbf{X}_d^i - \hat{\mathbf{X}}^i]^T \times [\mathbf{X}_d^i - \hat{\mathbf{X}}^i]}, \tag{3.21}$$

where i is the index of desired configuration and $i = \{1, \cdots, 20\}$. Given i, $\mathbf{X}_d$ and $\hat{\mathbf{X}}$ represent the desired configuration and the successive tracked configuration.

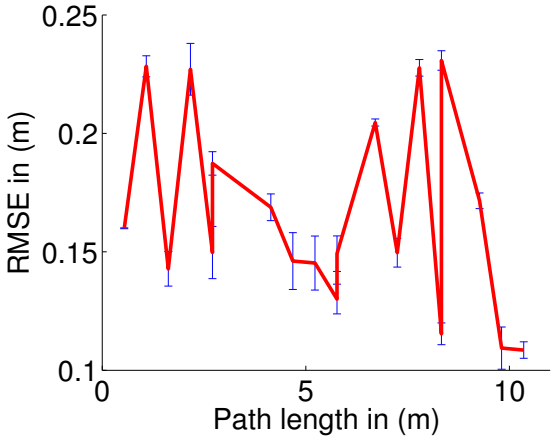
Fig. 3.5(a), shows the sampled trajectory for each experiment. The sampled trajectory was explored 4 times to determine the localization uncertainties shown in Fig. 3.5(b). The overview of experiment results were shown in Table 3.1. It took $104.13s$ to traverse the sampled path of $10.35m$. We observed maximum $0.23m$ RMSE error while traversing 20 waypoints. The maximum variance of RMSE from 4 experiments was noticed $0.012m$. The video available in the link `https://youtube/TUIh5ZLXM38` presents the demonstration of correspondent experiments.

3.4 Summary

This chapter provides our basic assumptions, and the brief explanation of mathematical tools which will be used in following chapters. Furthermore, we also implement a vision-based quadrotor model that can autonomously execute a given trajectory. The



(a) Sampled Trajectory



(b) Localization uncertainties

Figure 3.5: **Localization uncertainties:** Localization uncertainties were evaluated in terms of RMSE metric. The Sampled trajectory shown in (a) was traveled 4 times to compute the RMSE with variance shown in (b).

system does not rely on external positioning - GPS or motion capturing systems, as sensing. We describe not only the hardware platform and the software of our system but also detail the state estimation and the trajectory control. We report the experiments results with the video which is publicly available to the robotics community. Throughout the result we have shown that the perfect localization assumption is valid, since the error is almost negligible.

Chapter 4

Hotspot Localization

In this work, we take the hotspot seeking problem as the information gathering problem considering a threshold value to terminate the exploration. Focusing on the area coverage issues, we generate a regularly spaced hexagonal grid of sampling points. After localizing the hotspot, we propose a loop-closing path, allowing the UAV to return back to its initial position while visiting the informative locations.

Fig. 7.2 shows all the necessary steps for the proposed system.

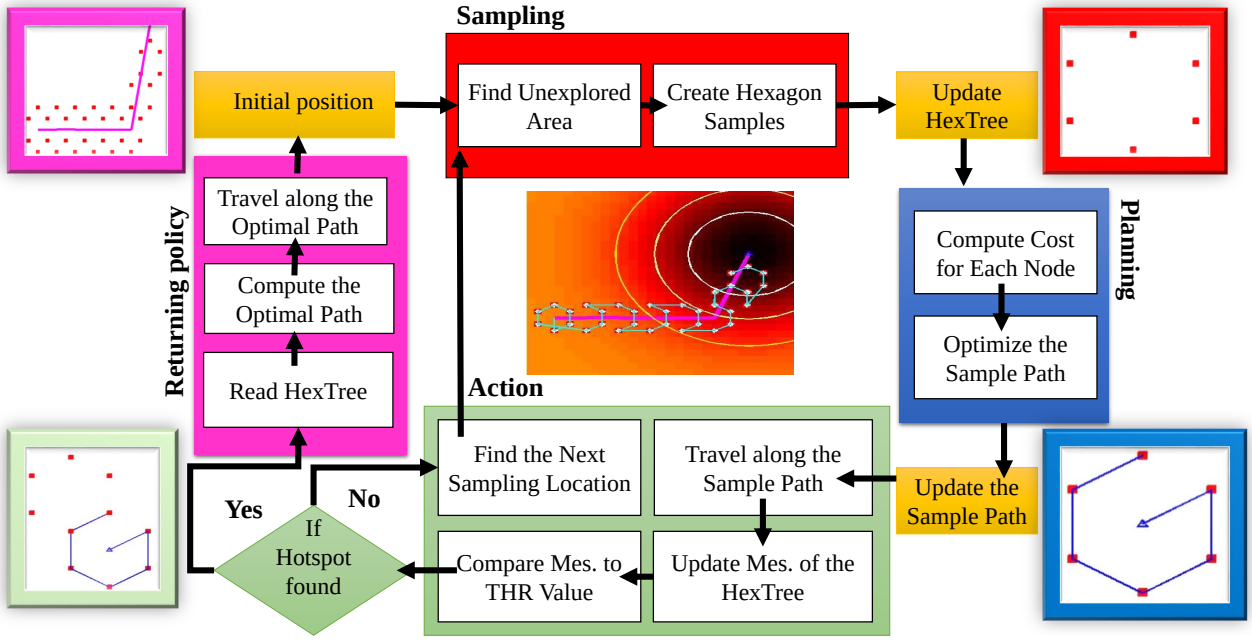


Figure 4.1: **System Overview:** The hotspot localization is performed by sampling, planning, and action phases. It is noteworthy that the UAV not only finds the hotspot but also returns to the initial position while visiting informative locations.

4.1 Problem formulation

A hotspot can be generated by multiple sources with unknown strength in a 2D area. Let $\mathbf{x_H} = \langle x_h, z_h \rangle$ denote the hotspot, where the postion of it is given by x_h and the expected measurement is a positive real number, $z_h \in \mathbb{R}_+$. A radiation field generated by multiple sources can be characterized using Gaussian Mixture Model [35]. Therefore, $\mathbf{x_H}$ can be remotely traced by the radiation sensors mounted on the UAV. For simplicity and without loss the generality, we assume that the measurement of the hotspot, z, from a remote location, x, can be given by

$$z\left(x\right) = z_h \exp\left(-\frac{||x - x_h||}{\sigma}\right) \tag{4.1}$$

where σ is the spread of the measurement and z_h is the strength of hotspot. If we assume that measurements are spatially distributed throughout the environment, then the hotspot seeking in a radiation field is somewhat similar to the informative path planning, where the planner queries the path subject to the maximal information gain, I. Let, at time t, the entropy of previously gathered measurements be $H(z_{1:t})$ and the entropy of measurements given the location x_t be $H(z_{1:t}|x_t)$. The information gain is then computed by the absolute difference of these entropies as follows

$$I := abs\left(H(z_{1:t}) - H(z_{1:t}|x_t)\right). \tag{4.2}$$

Using this metric, one of the efficient way for path planning is random sampling based approach [4]. In [4], information is gathered in two folds. Firstly, a number of sample locations are generated in a random manner and then measurements are gathered after traveling each of them. Let $\mathbf{n}$ be the set of samples at time t such that $\mathbf{n}_t = \{1, 2, ..., n\}$. A tree τ is then constructed by iteratively sampling in neighbor locations. Thus, the goal of an informative path planner at time t is to find a maximal informative location around the neighbors in such a way that

$$x_t^* = \underset{i \in \forall \mathbf{n} \in \tau}{\arg\max} \;\; abs\left(H(z_{1:t}) - H(z_{1:t}|x_t^i)\right). \tag{4.3}$$

If the sampling budget time T is given, the path $\mathcal{P}$ is then generated by assimilating all the local best locations given by

$$\mathcal{P} = \bigcup_{t}^{T} \{x_t^*\} \tag{4.4}$$

However, the hotspot seeking algorithm is considered as a goal oriented problem whereas in [4] the informative path planning is basically a goal free problem. An informative path planner can then find the hotspot by simply adding a termination condition, z_h,

such that

$$\mathcal{P}^* = \bigcup_{t}^{T} \{x_t^*\} \; s.t. \, z\left(x_t\right) < z_h \tag{4.5}$$

The problem of hotspot seeking becomes complicated when the information metrics for each sample location does not vary significantly at each sampling step. In that situation, we define the area coverage ratio as the performance index of the planner. We assume that the target area can be fully characterized by sampling the finite number of spatially distributed locations. Given a target area, $\mathcal{A} \leftarrow \forall \{x\}$, let the sampled locations be denoted by the set, $\mathcal{D} \leftarrow \cup\{\exists x \in \mathcal{A}\}$. The UAV will visit only unexplored area and when $\mathcal{A} \backslash \mathcal{D} \leftarrow \{\}$, the area would be fully covered. Let $\mathcal{D}_{t+1}^i$ be the prediction over $\mathcal{D}_{t+1}^i$ given the next sampling location x_t^i, and $c\left(x_t^i\right) \triangleq \frac{|\mathcal{D}_{t+1}^i|}{|\mathcal{A}|}$ be the area coverage ratio at exploration time, t. Since expensive explorations are required to gather the measurement attributes, z_t, we want to minimize the area coverage ratio at each sampling step t. Thus, our desired hotspot directed sampling path can be expressed as follows

$$\mathcal{P}^* = \min_{c(x_t)} \max_{I(x_t)} \sum\nolimits_t \left(c\left(x_t\right) + \eta I\left(x_t\right)\right) s.t. \, z\left(x_t\right) < z_h \tag{4.6}$$

where η is normalizing constant. Although a sampling path is the necessary condition to localize an unknown hotspot, it is observed that all the sampled locations are not informative. Thus, when the UAV has to return back its initial location, it can visit only to the most informative locations to make an efficient tour. In this way, we can then find the loop closure path by visiting only to the most informative locations. Let W be the total information gain for the loop closure path, which can be computed by visiting all the sampled locations. Since we already know all the sample locations $x \in \mathcal{D}$ and their corresponding information $I\left(x\right)$, the informativeness of each location can be evaluated by a weight, w, as follows

$$w(x) \triangleq \frac{I\left(x\right)}{dist\left(x, x_0\right)} \tag{4.7}$$

where x_0 is the initial position of UAV and $dist$ is the function that computes Eucledian distance. Therefore, the optimal returning path can be obtained through avoiding less informative locations given by

$$\mathcal{P}_{opt}^* = \max \sum_{x \in \mathcal{D}} I\left(x\right) x \;\; s.t. \sum_{x \in \mathcal{D}} w(x) x \; <= W \tag{4.8}$$

In summary, the goal of this thesis is to answer the following question- *given a termination threshold z_h of an unknown hotspot, $\mathbf{x_H}$; how to generate a tour path, $\langle \mathcal{P}^*, \mathcal{P}_{opt}^* \rangle$, to quickly localize the hotspot position x_t, where $z\left(x_t\right) > z_h$?*

4.2 Algorithm Descriptions

We now discuss the sampling strategy that generates the UAV trajectory toward the hotspot, minimizing the area coverage ratio. The key idea is to use the samples over a set of hexagonal grids and to build a tree of possible trajectories by extending candidate trajectories toward the sampled points. The hexagonal grid has two benefits: it covers more directions compared to the rectangular grid, and is the most efficient shape that can tile the 2D plane [88].

4.2.1 Local field Sampling

We propose a HexTree based sampling strategy inspired from the RIG-tree structure [4], where vertices in the graph represent a tuple of location, cost, and information. Unlike the RIG-tree, a HexTree has a fixed number (six) of children nodes, q_{hex}, centering at a (virtual) parent node v_t. As the name implies, the children nodes are hexagonally distributed, and each of them is separated from its neighbors by the same interval a, and the adjacent parents share common children nodes depending on their locations. The root of the tree represents the initial position. During the sampling period, the robot visits only the children nodes and then the planning decision is stored in the parent node. Thus, each parent node contains the information of the local best child node.

Algorithm 2 HexTree formulation

Require: $\mathcal{A}$, x_0
$Tree$
$Tree.AddRoot(x_0)$
$v_t \leftarrow x_0$
while not converged to hotspot **do**
  $q_{hex} \leftarrow HexagonalSampling(\mathcal{A}, v_t)$
  $q_{sort} \leftarrow GetSamplePath(q_{hex})$
  $q_{best} \leftarrow FindMaxInfoEdgeOnTree(Tree, q_{sort})$
  $v_{t+1} \leftarrow GetNewParent(\mathcal{A}, Tree, q_{hex}, q_{best})$
  for all $i \in \mathbf{n}_t$ **do**
    if $z_t^i > z_h$ **then**
      return $OptimalReturnPath\,(tree)$
    end if
  end for
end while

4.2.2 Sampling path generation

The tree structure is now under a favorable condition to reduce the UAV exploration using radiation field properties. We can create an incremental version of Hamiltonian path to visit each child location, $x_t^i \in q_{hex}$, at most one time and avoid redundant visits to the same location. Instead of n number of random points, the HexTree constraints sample points into six at sampling time t, and that some of them are shared. Therefore, we can optimize the sample path to only unexplored locations. In order to generate a Hamiltonian path to the current UAV location, first, we assign a global index set J with the index function ind. ind generates a unique index for each node mapping the node from the position domain $x_t^i \in \mathbb{R}$ into the index domain $ind(x_t^i) \in \mathbb{N}$. We store each visited node into $J = \cup\{ind(x_t^i)\}$. Secondly, we compute the traveling cost between the UAV and each new node position given by

$$q_t^i.cost = dist(x_t^i, x_0)_{ind(x_t^i) \notin J} \tag{4.9}$$

Once we compute the cost of each node, we sort all the candidate nodes using the Hamiltonian path algorithm [89]. Thus, we find the optimal sequence to sample the local hexagonal grid. Note that, even though we sample the area using a regular hexagonal grid, the sample path does not need to follow such a pattern. At each step, the UAV exploration is ended with a candidate child location and the subsequent sample path is generated with the evolved location.

Algorithm 3 Sampling path generation

Require: q_{hex}
1: **for** $\forall x_t^i \in q_{hex}$ **do**
2: **if** $ind(x_t^i) \notin J$ **then**
3: $q_t^i.cost \leftarrow dist(x_t^i, x_0)$
4: **else**
5: $q_t^i.cost \leftarrow \infty$
6: **end if**
7: **end for**
8: $q_{sort} \leftarrow HamiltonianPath(q_{hex})$
9: **return** q_{sort}

4.2.3 The next best parent

Given the sample path defined by Alg. 3, the robot travels each of the node location at most one time as similar to Fig. 4.2 and gathers the measurement attributes. Based on the gathered information, the next best parent location is computed using the following steps- firstly, we assign the virtual edges for each sample location deterministically, for

instance, the virtual edges $\{BF, EF, EA, ...\}$ in Fig. 4.2 represent the nodes $\{B, F, E, ..\}$. Secondly, we compute the information gain for each node using the Eqn. 4.6. Thirdly, the edge which has maximum information gain, we find the neighbor edges. For instance, the edges HB and EF are the neighbor edges of BF in Fig. 4.2. Finally, we extend the neighbor edges in such a way that coincidence a point, G in Fig. 4.2, which is the next best parent location. Note that, G is a virtual point where the robot does not visit. It is only computed to generate the next sample locations in a hexagonal manner.

4.2.4 The optimal return path

While the UAV iteratively seeks the hotspot location, all the sampled locations are not likely to be informative due to the fixed sample interval. We therefore generate an optimal return path using the dynamic programming based Knapsack algorithm [90]. Specifically, we constrain the travel distance while maximizing the information gain. Since the HexTree already tracked all the information through the sampling step, we convert it to the Knapsack variables. The return path abandons those locations where the measurement does not add much important information to the field. The local field was represented by the child node having the maximum information gain, which was also stored in the virtual parent node. Neglecting the distance between such a child node and its parent node, we can easily generate the return path based on the parent node locations only.

4.3 Optimality Analysis

Our approach is to tile the hexagon through the sample nodes using the area coverage strategy. Since the tiling process involves node sharing, the redundant visit can be avoided by generating the Hamiltonian path. However, when the robot has to avoid the shared nodes, there can be an increase in the path length. We will now give an upper bound on this path length.

Definition 4.3.1. *Let's assume that 'a' is the step length which is required to travel two consecutive children nodes in a parent grid and 'n' is the total number of parents in the tree. An online algorithm solving the navigation problem is said to be optimal if the total path length is bounded by a factor* $\left(5 + \sqrt{3}\right) na$

We first state a number of modest assumptions that are required for this analysis. Next, we place a bound on the sampled path by finding the proofs of two essential theorems.

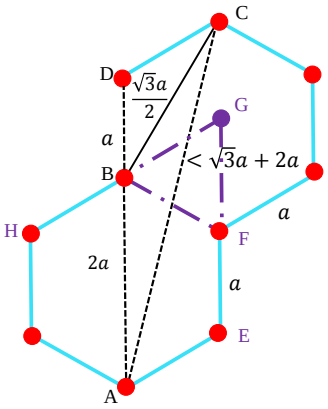


Figure 4.2: **HexTree Sampling Pattern:** The red dots are the sampled locations and the cyan lines are the sampled path. The purple lines are drawn to determine the next parent (virtual) location denoted by G.

Assumption 4.3.1. *There exists at least one new child node in each parent node to extend the tree towards the goal position.*

This assumption requires that some free space is available to create a sample parent. Since two consecutive sample parents share at least one edge, to make a valid sample patent at least one new vertex is required.

Assumption 4.3.2. *Starting with an arbitrary location, the robot travels to each child node of a sample parent in a Hamiltonian path manner.*

We want to optimize the total length of exploration time. Thus, this assumption leads the navigation problem to an optimal solution. Since the adjacent parents are sharing at least one edge- which means at least two children nodes, traveling to all the children at most one time is helpful to avoid the redundant visits.

Theorem 4.3.1. *If 'n' number of hexagons are required to reach the goal position, there exists at least $5n$ edges which are required to generate a Hamiltonian path*

Proof. Given a hexagonal grid, initially, the robot can travel at most $6-1=5$ vertices to make a Hamiltonian path. If this pattern follows, then it is straightforward to conclude that $5n$ edges are required in total. However, since two adjacent hexagonal grids share at least one edge, the robot skips 2 vertices while travailing to all the children nodes. From Fig. 4.2, we can understand that the number of edges which is required to extend the Hamiltonian path for the adjacent grid is $6-2=4$. However, since the robot needs to travel at least distance a to reach the adjacent grid, which is equivalent to the length of an edge, we can then compute the total edges as follow

$$4n + n = 5n \tag{4.10}$$

□

Theorem 4.3.2. *If 'n' number of parents are required to extend the tree onto the goal position and 'a' is the distance between the consecutive neighbors, the sampled path for the HexTree can be extended upto* $a\sqrt{3}(n-1)$.

Proof. In the worst case, the robot travels to the farther child node of the neighbor parent and the neighbor parent shares only two common children. Fig. 4.2 shows the worst case scenario, where the robot travels to the farthest child node. Let's assume that the robot is located at node A and the goal node is denoted as B in the Fig. 4.2. In order to compute the worst case path distance, AC, we draw two imaginary edges from A to B and B to C. Since a regular hexagon comprises of six equilateral triangles, the length of BC can be computed from the triangle BDC as follows

$$BC = 2\sqrt{(a^2 - a^2/4)} = a\sqrt{3} \tag{4.11}$$

And it is obvious from the Fig. 4.2 that the length of AB is the diameter of the hexagon, computed by

$$AB = 2a \tag{4.12}$$

Thus, using the triangle inequality theorem for the triangle ABC, we can state that $AB + BC > AC$. As a result we can compute the upper bound of AC from the Eqn. 4.12 and the Eqn. 4.11 as follows

$$AC < (a\sqrt{3} + 2a) \tag{4.13}$$

However, as we know from the definition that the minimum travel distance between two consecutive children nodes is a and in the worst case scenario two consecutive parents are shared at least one edge which length is also a, thus, the maximum length of AC can be exceeded upto

$$\Delta AC = a\sqrt{3} + 2a - a - a = a\sqrt{3} \tag{4.14}$$

As we can see from the Fig. 4.2, we need at least two hexagons to compute the length of AC, therefore, the worst case path could be extended by the n number of hexagons as follows

$$\sum\nolimits_{n-1} \Delta AC = (n-1)\Delta AC = a\sqrt{3}(n-1) \tag{4.15}$$

□

Combining the theorems and the above equations, we can compute the final bound

Table 4.1: Overview of parameters

Parameter	Value	Description
$\mathcal{A}$	$30m \times 30m$	Target area size
$\mathcal{H}_{str}$	3000	Strength of hotspot
$\mathcal{H}_{pos}$	[25 20]	Position of hotspot
σ	[150 150]	Spreading matrix
a	$1m$	Sample step length
ξ	2900	Termination threshold

Table 4.2: Algorithm performance

	RIG Tree	HexTree
# iterations	55	7
# samples	55	42
path len.(m)	24.849	7.865
time (s)	104.364	43.256
std. dist. (m)	10.416	0
Max path len. (km)	∞	2.33

of the sampled path as follows

$$\sum_{n-1} \Delta AC + 5na < a\sqrt{3}(n-1) + 5na \approx (5+\sqrt{3})na \tag{4.16}$$

4.4 Simulation Results

We present results of numerical experiments with the HexTree planner compared to the RIG-tree planner. Our return path is also compared to the path smoothing algorithm. Table I presents an overview of parameters used for the experiments. Each algorithm is implemented in MATLAB on a PC with a $3.40GHz$ Intel(R) Core(TM) i7 processor and $8.0GB$ RAM.

4.4.1 Compared Strategies

The RIG-tree planner is in principle capable of finding the hotspot, which is asymptotically optimal. However, it takes a very long time to converge to the hotspot. For a single UAV exploring over a large area, the HexTree planner is even faster than the RIG-tree planner. To compare the performance of both planners, we measure the total length of sampled path and the time to converge with the same initial position, step size, and termination condition.

Fig. 6.8 shows the difference between the HexTree and RIGtree planners. The Hex-

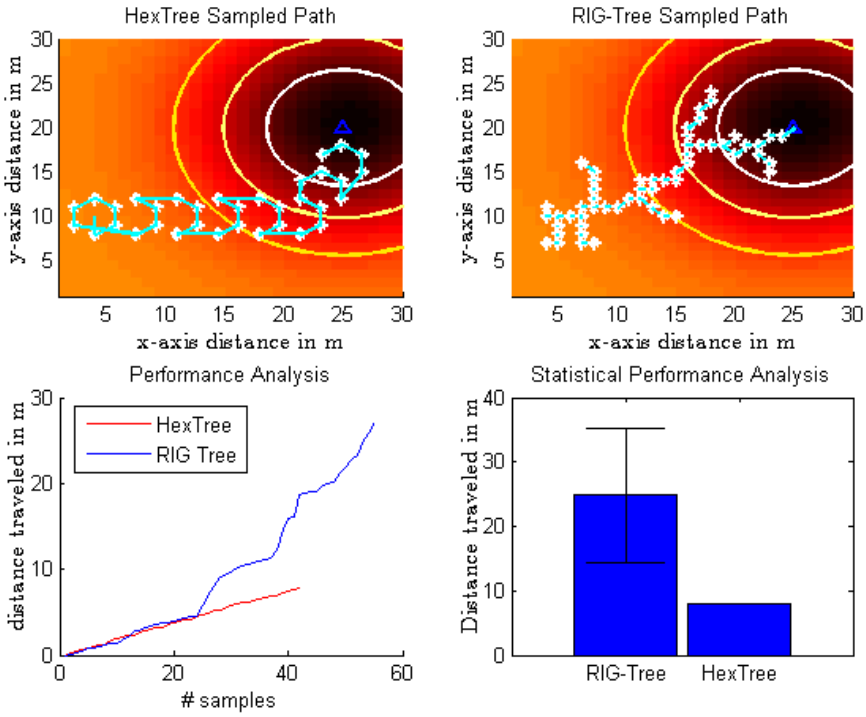


Figure 4.3: **Performance comparison:** White dots are the sample locations, cyan lines are the sampled path, the blue triangle is the hotspot position, the baseline colored map is the radiation distribution. A UAV needs to sequentially explore each location to gather the information. The RIG tree does not have sequential path generation characteristics.

Tree sampled path was restricted by the hexagons, and only 7 iterations were required to converge to the hotspot, while 55 iterations were required for the RIG-tree. Furthermore, since the HexTree sampled path was sequentially optimized, we can see from Table 4.2 that the HexTree traversed only $7.865m$ distance to find the hotspot, whereas the RIG-tree traversed $24.849m$. Since the RIGtree randomly sampled the area, it required more sample points (55) and longer time ($104.364s$). On the other hand, the HexTree clearly outperformed the RIG-tree in terms of the sample point (42) and the convergence time ($43.256s$). The total distance required to reach the hotspot with the RIG-tree varied significantly across 10 experiments, while the HexTree remained constant. In case of the absence of hotspot, the RIG-tree could not find any condition for which sampling will be terminated, resulting in a sampled path of infinite length. On the other hand, the HexTree needed only 346 iterations to completely cover the $900m^2$ area, resulting in $2.33km$ sampled path to reach a termination point.

4.4.2 Path Smoothing Vs. Optimal Return Path

The sampled path may contain locations that lead to redundant visits in view of information gathering. Optimizing the return path is therefore an essential step similar to the RRT [37] path smoothing. Our focus is to find the most informative locations considering the travel cost. As the RIG-tree does not have the return path, we compare our return path with the RRT path smoothing. We have performed 3 simulations with different initial UAV positions and hotspot locations as shown in Fig. 4. Since the path smoothing finds the shortest path to the initial position, the return path for the RIG-tree looks more like a straight line and cannot find any informative locations from the sampled path. Meanwhile, the proposed path finds several locations considered as the most informative locations satisfying the travel distance constraint.

4.5 Summary

We address the problem of hotspot seeking in an unknown radiation field using an Unmanned Aerial Vehicle (UAV) with limited resources. For on-the-spot investigation of accidental radiation releases, without *a priori* knowledge on the whereabouts of the source of radiation substances leakages, it is very difficult to navigate and return a UAV for fast hotpost localization. We propose a novel Hexagonal Tree (HexTree) based sampling algorithm to find such an optimal tour path based on the appropriate measurement locations. We make a realistic assumption on the environment, theoretically analyze the optimality of proposed algorithm, and numerically compare the performance with the existing method. The proposed algorithm gives faster convergence to

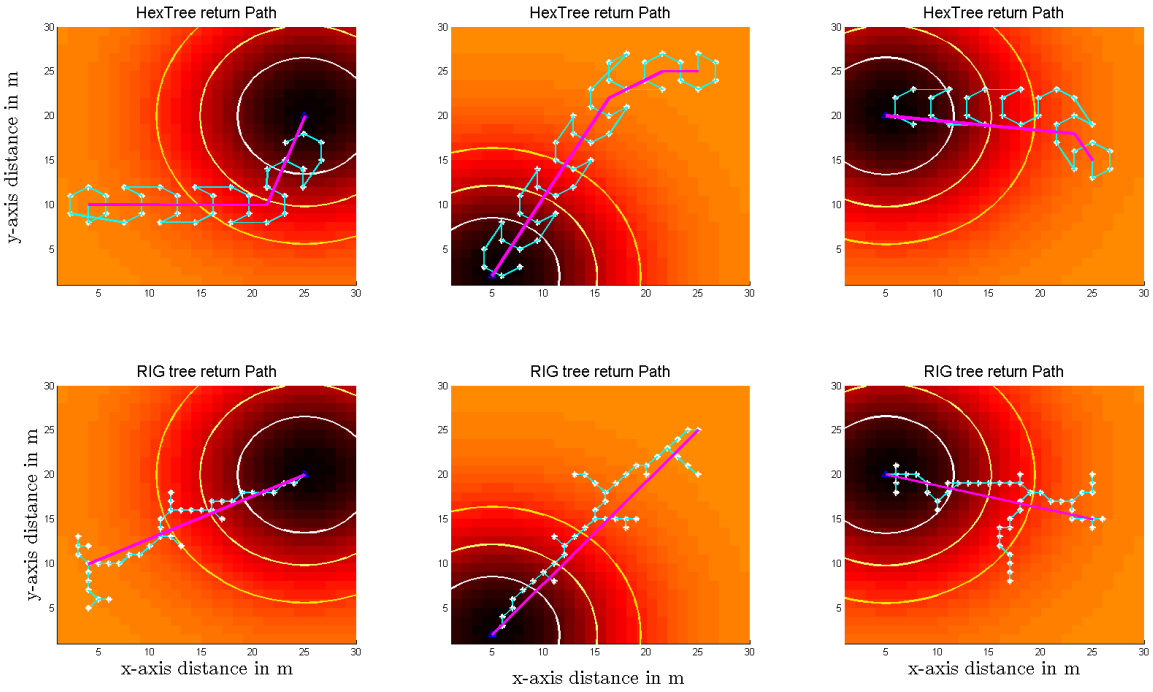


Figure 4.4: **Return path generation:** HexTree return path is generated while visiting informative locations. On the other hand, the path smoothing algorithm finds the shortest path to the initial location subject to distance constraints.

the hotspot, an optimal exploration termination condition, and more informative locations while returning to the initial position than conventional random sampling based exploration and path smoothing algorithms.

Chapter 5

Boundary Estimation

5.1 Problem Formulation

In this work, we want a single UAV to have the ability to estimate an unknown boundary. To enable this robot to estimate such boundary which is defined by threshold β, it needs to sample along the boundary to gather measurements in a point-wise fashion. The objective is to track spatial sampling points, $\mathbf{S}$, and augment these with a smoothing function to estimate the desired boundary. Thus, the problem we want to solve here stated as follows:

Given a threshold value β for an unknown environment that may include non-concave regions and the measurements subject to noise w_t, how to accurately track a closed boundary set $\mathbf{S}$ *in a finite time limit.*

Similar to [15, 12, 16], we also assume that the boundary is described by a smooth, regular, simple, closed curve.

5.1.1 Field Characterization

Our focus is to find a region within the environment where there is a phenomenon delimited by a perimeter. This region of interest $\Omega \in \mathbb{R}^2$ is a finite set and enclosed by a boundary. The measurement of a location in such field defined as a map

$$z(x) : \mathbb{R}^2 \to \mathbb{R}, \tag{5.1}$$

that evaluates the strength of the phenomenon at the point x, and expressed in intensity unit.

Definition 5.1.1 (Region of interest (ROI))**.** *The region of interest is the collection of points in the environment where the measurement $z(x)$ is greater than some threshold value β, i.e. the set $\{x \in \Omega | z(x) > \beta\}$.*

Our boundary of interest $\mathbf{S}$ is a simple closed curve that represents the perimeter of ROI. Since we want to minimize the robot's exploration, the ROI can be determined by tracking the boundary line $\mathbf{S}$ only.

Definition 5.1.2 (Boundary line)**.** *The closed curve is said to be boundary line if it represents a level set* $\mathbf{S}$ *such that for each point* $x \in \Omega$*, the measurement satisfies threshold* β*, i.e. the set*

$$\mathbf{S} := \{x \in \Omega | z(x) = \beta\} . \tag{5.2}$$

Note that it is common to assume that the measurements varies linearly in the vicinity of boundary line [23, 16]. Therefore, a linear controller is sufficient enough to sample such kind of boundaries.

5.1.2 Spatial Sampling

To sample the environmental boundary $\mathbf{S}$, we use a single UAV to gather the measurement in a point-wise manner. We assume that only 2D Euclidean trajectory is sufficient enough to estimate $\mathbf{S}$. Therefore, throughout this chapter, the UAV's position represented by x, i.e. $x \in \mathbb{R}^2$, and it expressed in polar coordinate system.

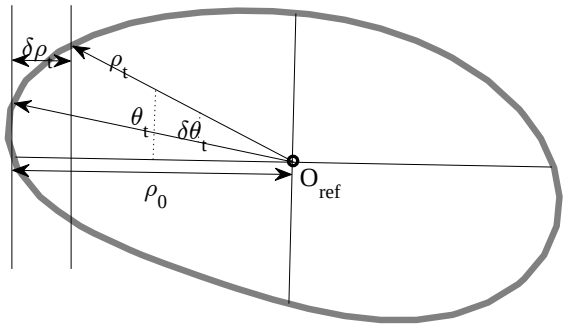


Figure 5.1: **Boundary Estimation:** A environmental boundary is estimated by varying the polar radius ρ_t and angle θ_t w.r.t the reference origin O_{ref}

Fig. 5.1 shows the key concept to make a closed boundary, which is similar to our previous work [**?**]. The overall process is in two folds: Firstly, we transform the origin of the polar coordinate to the initial robot's neighborhood location denoted by O_{ref}. It is a random location inside the boundary, where $z(O_{ref}) > \beta$. Secondly, by continuously varying the polar radius ρ and polar angle θ, we find the level set curve $\mathbf{S}$.

5.1.3 Controller Synthesis

Let at time t the robot 2D-coordinate be represented by the polar angular position θ_t and radius ρ_t such that $x := [\rho_t, \theta_t]$. To track the $\mathbf{S}$ at each time step t, a point-2-point

controller predicts a relative location $u : \mathbb{R}_+ \times [0, 1] \rightarrow \mathbb{R}^2$ as follows

$$u_t(\delta\rho_t, \delta\theta_t) = \delta\rho_t \times \begin{bmatrix} \cos\delta\theta_t \\ \sin\delta\theta_t, \end{bmatrix} \tag{5.3}$$

where $\delta\rho_t$ and $\delta\theta_t$ are predicted increments of radius and angle respectively. These increments are subjected to the measurement of the current location $z(x_t)$. The robot is assumed to be able to localize itself within the environment with a negligible on the boundary estimation.

After executing every motion, the distance traveled by the robot to track the $\mathbf{S}_i$ can be defined as the curvature of the boundary, denoted by κ. Let l be the length of the curvature when the robot finishes its tracking by returning back to the initial position. To approximate a boundary, the robot has to explore the vicinity of the boundary of interest in a such way that satisfies the Eqn. (7.11). Since the measurement affected by sensor noise, we can then explain the robot's observation as follows

$$h(x_t) = z(x_t) + w_t, \tag{5.4}$$

where w_t is the Gaussian white noise with zero mean. If *apriori* initial approximation of $\mathbf{S}$ is available, the robot can then estimate the $\mathbf{S}$ by n vertices polygon P_n^* by measuring the Lebesgue similarity δ^s as follows

$$\delta^s\left(\mathbf{S}, P_n^*\right) = \int_0^{2\pi} \rho\left(\theta\right)^\alpha d\theta = \int_{l\in\mathbf{S}} \kappa(l)^{(1-\alpha)} dl \tag{5.5}$$

where $\alpha > 0$ which represents the polynomial degree and $\kappa := \rho^{-1}$. To construct the best P_n^*, Melure and Vitale [91] consider $\alpha = 3$. However, there are several flaws in the polynomial approximations such that

- if an initial approximation of the boundary is not given, determining the polynomial degree α is troublesome.
- in order to approximate an area $\mathcal{A}_i$, the $n \in P_n^*$ should be sufficiently large. As a result, for an unknown and arbitrary area $\mathcal{A}_i$, it is then difficult to deterministically choose the value of n.
- it requires to store a large number of variables, resulting in a substantial increase of memories as well as computational expenses, which restricts the possibility to perform a wide-scale computational experiment.

On the other hand, removing the assumption on the initial estimation, the simplest tracking algorithm for a single robot is the bang-bang controller, where the robot keeps

changing its direction subject to threshold β. For instance, when $h(x_t) < \beta$ it moves inside direction and in outside direction when $h(x_t) > \beta$. This controller works well except for a few drawbacks.

- with large $\delta\theta_t$, the tracking becomes very inefficient [16].
- since the robot's observation is affected by noise $w(t)$, with a large $w(t)$, it may turn the wrong way and fail to track the boundary.
- when the boundary has narrow bottlenecks, the controller fails to cover it [92].

Incorporating the filter for denoising measurements is standard practice for navigation tasks. However, the crossing angle correction is less common within robotics. To limit the crossing angle, Jin et al. [16] proposed the following modification

$$\delta\theta_t := sgn(h(x_t) - \beta)\,(t \cdot \tilde{\omega} - 2\theta_0)\,/2, \tag{5.6}$$

where θ_0 is a preset reference, and $\tilde{\omega}$ is the angular velocity of robot. In recent year, Matveev et al. [23] proposed a side way controller in a dynamic environment with following modification

$$\delta\theta_t := sgn\left(\dot{h(x_t)} - \chi[h(x_t) - \beta]\right), \tag{5.7}$$

where χ is a linear function with a preset saturation and $\dot{h(x_t)} = \frac{h(x_t)}{dt}$. However, tuning such preset parameters for a controller are troublesome, especially in an unknown environment.

5.2 Algorithm Description

5.2.1 State Estimation and Prediction

Let us denote the robot location at time t as $x_t = (\rho_t, \theta_t)$ with the respect to O_{ref}. When the robot visits a location x_t, it receives the field measurement for the corresponding location denoted by $z(x_t)$. The robot motion is generated by a controller subject to somewhat predictive function given by as follows

$$f(x_t, u_t) = x_t + u_t = \begin{bmatrix} \rho_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \delta\rho_t \\ \delta\theta_t \end{bmatrix} \tag{5.8}$$

where $u_t = [\delta\rho_t, \delta\theta_t]^T$ is the controller output that represents the next relative location based on the observation on x_t location given by

$$h(x_t) = \begin{bmatrix} \rho_t \\ \theta_t \\ z(x_t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_t \end{bmatrix} \tag{5.9}$$

where w_t is Gaussian white noise. The goal of boundary estimation is to infer the robot location based on knowledge of the control actions and observations. The level set curve is generated by accumulating all the robot's locations. Since the robot's observation is affected by the random noise, we use the Extended Kalman Filter (EKF) to make a better prediction.

5.2.2 Controller Design

In this section we describe the process for designing a controller for estimating environmental boundary while respecting the constraints of a single robot exploration. For estimating the environmental boundary, the robot needs to explore the environment and exchanges measurements. Given the state x_t, a controller makes use of the corresponding measurement $z(x_t)$ in order to predict the next target state x_{t+1}. We use a PID controller where the control signal u_t is given by

$$u_t = \begin{bmatrix} k_P \cdot e + k_I \cdot \int e dt + k_D \cdot \dot{e} \\ \delta\theta_t \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} \mathbf{e}(i)\mathbf{k}(i) \\ \delta\theta_t \end{bmatrix}, \tag{5.10}$$

where $\mathbf{e} = [e, \int e dt, \dot{e}]^T$ and $\mathbf{k} = [k_P, k_I, k_D]^T$ are the vectors contain the proportional, integration and derivative errors and gain respectively. Given the current measurement $z(x_t)$, the relative location of the robot can be found by computing the error metrics as follows

$$e = \beta - z(x_t), \quad \dot{e} = \frac{de}{dt}. \tag{5.11}$$

We can then reformulate the state prediction used in Eqn. (5.8) by plugin the variable u_t from Eqn. (5.10) as follows

$$f(x_t, u_t) = \begin{bmatrix} \rho_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} \rho_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^{3} \mathbf{e}(i)\mathbf{k}(i) \\ \delta\theta_t. \end{bmatrix} \tag{5.12}$$

This integration over the state naturally takes into consideration the error metric of the next relative location is expected to low: if the robot samples very close to the boundary then e will be low as most exploration steps, while if the sampling location is far from the boundary the e will be larger. However, it does not take into account the angle

correction aspects, an issue for exploration over non-concave boundaries which may lead to falsely determining the boundary, as noted in [16].

5.2.3 Adaptive Crossing Angle Correction

In order to accurately track the environmental boundary, the non-concave regions must be crossed at fine angle correction. The main idea is that initially a coarse angle increment $\delta\theta$ is used and is reined only in areas that are likely to contain a non-concave boundary. If the error metric in Eqn. (5.10) turns out to be a large value, the procedure can then be activated. The basic operation of this adaptive angle correction can be explained as follows

$$\delta\theta_{t+1} = abs\left(\delta\theta_t - e \cdot \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho_{t+1}^2}\right)\right). \tag{5.13}$$

For a non-concave boundary the error metric e is higher, resulting in large ρ_{t+1} value. With a large ρ_{t+1}, the Eqn. (5.13) returns a small $\delta\theta_t$ value. As a result, the robot adds more exploration with fine angular increment in non-concave surface. Thus, we can rewrite the Eqn. (5.12) as follows

$$f(x_t, u_t) = \begin{bmatrix}\rho_{t+1}\\ \theta_{t+1}\end{bmatrix} = \begin{bmatrix}\rho_t\\ \theta_t\end{bmatrix} + \begin{bmatrix}\sum_{i=1}^{3}\mathbf{e}(i)\mathbf{k}(i)\\ abs\left(\delta\theta_t - e \cdot \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho_{t+1}^2}\right)\right).\end{bmatrix} \tag{5.14}$$

Eqn. (5.14) is an accurate prediction to our true boundary estimation objective. The intuition behind this is that the robot should travel the nonconcave regions with fine angular increments to better estimate the boundary, while the concave regions can be travel with coarse angular increment. Using this reasoning it is simple to adapt the definition of angle correction used in [23, 16]. The angle correction in [23, 16] is chosen deterministically, which limits wide application; while our approach offers an adaptability of how to best explore any random non-concave regions in an unknown boundary.

5.2.4 Boundary Tracking Algorithm

Single robot's boundary tracking algorithm iteratively chooses a destination target to minimize the expected error metric of the predicted location based on the current estimate. A tour through these targets estimate the length of boundary as follows

$$\int_t \sum_{i=1}^{3}\mathbf{e}(i)\mathbf{k}(i)abs\left(\delta\theta_t - \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho_{t+1}^2}\right)e\right)\delta t \approx \int_{l\in\mathbf{S}}\kappa(l)dl. \tag{5.15}$$

We can find a closed loop path in solving Eqn. (5.15) by replacing the boundary limit from time domain to angular domain. The following theorem establishes a convergence bound for Boundary tracking algorithm 1 in terms of finite time termination condition.

Theorem 5.2.1 (Convergence theorem)**.** *When the UAV samples an unknown environment, with prediction model specified by (5.14) and control law specified by (5.10), we have:*

$$\lim_{t\to\infty} f(x_t, u_t) \to x_0. \tag{5.16}$$

Proof. Let, at $t = 0$ the robot's position be $x_0 = [\rho_0, \theta_0]^T$. In a closed path, the robot's final location is the same as initial location. The final location can then be found when the robot will be $x_t = [\rho_0, 2\pi + \theta_0]^T$. Let assume that the following relationship holds

$$\int_{\theta_0}^{2\pi+\theta_0} \rho\delta\theta = \int_{\mathbf{S}} \kappa(l)dl. \tag{5.17}$$

Thus, we can rewrite Eqn (5.15) as follows

$$\begin{aligned} \int_t \sum_{i=1}^{3} \mathbf{e}(i)\mathbf{k}(i) \cdot abs\left(\delta\theta_t - e \cdot \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho_{t+1}^2}\right)\right)\delta t = \\ \int_{\theta_0}^{2\pi+\theta_0} \sum_{i=1}^{3} \mathbf{e}(i)\mathbf{k}(i) \cdot abs\left(\delta\theta_t - e \cdot \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho(\theta_{t+1})^2}\right)\right). \end{aligned} \tag{5.18}$$

Given $\rho_0 >> \delta\rho_t$ and $\theta_0 >> \delta\theta_t$, the requirements in Eqn. (5.18) met by all small enough $\delta\theta_t$. Since $abs\left(\delta\theta_t - e \cdot \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho(\theta_{t+1})^2}\right)\right) > 0$ in Eqn. (5.18), θ_t monotonically increases in every iteration. Thus, when $t \to \infty$, then $\exists\theta_t$ that goes to $\theta_0 + 2\pi$, resulting in termination with finite time limit. Then, the robot can reach its initial location x_0 for the angle $\theta_0 + 2\pi$ by overlooking the increment of $\delta\rho_t$, *i.e.* $\delta\rho(\theta_0 + 2\pi) \approx 0$.

□

The overall boundary tracking algorithm is explained in Alg. 1. It is important to note that this is not a traditional PID algorithm in [8], as the EKF based noise cancellation and adaptive angular correction features are incorporated into the target state estimation. As a result, this boundary tracking algorithm exhibits the robust and accurate estimation of an unknown environmental boundary so that the robot naturally minimize the length of regions where the estimation accuracy is poor.

5.3 Simulation Results

The three scenarios considered in the simulations shown in Fig. 5.2. The radiation fields are simulated in Gaussian Mixture Model (GMM) with two components. The

Algorithm 4 Boundary tracking

Require: initial state x_t
Ensure: Boundary $\mathbf{S}$

1: $\mathbf{S} \leftarrow \{x_t\}$
2: **while** $x_t(2) < 2\pi + x_t(2)$ **do**
3: Get measurement

$$z(x_t) = h(x_t)$$

4: Compute controller output

$$u_t = \begin{bmatrix} \sum_{i=1}^{3} \mathbf{e}(i)\mathbf{k}(i) \\ \delta\theta_t \end{bmatrix} = \begin{bmatrix} \rho_{t+1} \\ \delta\theta_t \end{bmatrix}$$

5: Perform angle correction

$$\delta\theta_{t+1} = abs\left(\delta\theta_t - e \cdot \operatorname{atan}\left(\frac{\delta\theta_t^2}{\rho_{t+1}^2}\right)\right)$$

6: Predict target state with EKF

$$x_{t+1} = f(x_t, u_t)$$

7: Update boundary

$$\mathbf{S} \leftarrow \cup\{x_{t+1}\}$$

8: **end while**

GMM satisfies the following equation

$$GMM(\Theta) = \sum_{i=1}^{2} \pi_i \mathcal{N}\left(\mu_i, \Sigma_i\right), \tag{5.19}$$

where each vector component is characterized by normal distributions $\mathcal{N}$ with weights π_i, means μ_i and covariance matrices Σ_i. The measurement of the field is represented by the random points distributed according to Eqn. (5.19). The robot sensing range is defined by a circle with radius $0.25m$. The measurement of a location is computed by the number of particles inside the robot's sensing range.

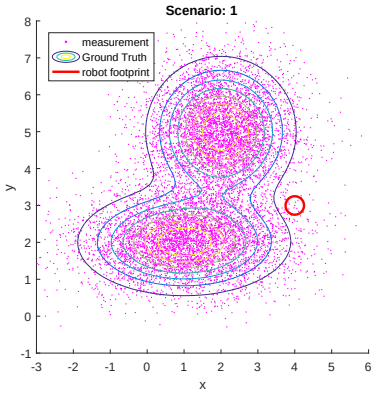
We have conducted 9 experiments in total for 3 different scenarios with 3 factors. For each scenario, the initial robot position was defined randomly. The measurement of the initial position was considered as the boundary threshold explained in Eqn. (7.11) such that $\beta = z(x_0)$. To sample the environment, the state prediction of the robot were considered by varing two factors as explained in Section 3: angle correction and noise cancellation. Each experiment was evaluated into two phases- exploration phase and estimation phase. The exploration phase corresponds to the robot traveling through its predicted location, without further reshaping it. The estimation phase corresponds to when the exploration phase is finished and a smooth boundary is generated over explored locations.

Fig. 5.2 depicts the scenarios considered in our experiments. All the parameters for GMM specified in sub captions of Fig. 5.2. The ground truth of the radiation field visualized by a set of blue contour lines, while the measurements of the field represented by pink dots. The robot's position denoted by the red circle. Given a robot location, the measurement of that location computed by the density of pink dots within the red circle. Since the origin of robot's coordinate could be any random point inside the contour lines. We did not explicitly show the origin location.

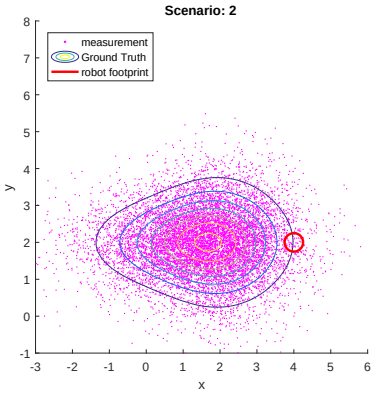
5.3.1 Exploration Phase

During this phase, the robot starts from an initial location and generates a closed path by tracking β. To generate the explored path, we perform 3 experiments for each scenario by varying 3 factors: 1) the robot travels its entire path with deterministic angle correction, measuring the sensory function $z(x_t)$ without any noise filter. 2) the robot travels its entire path with adaptive angle correction as explained in Section 3, but any noise filter. 3) the robot travels its entire path with the adaptive angle correction and the EKF as explained in Section 3. Since θ is bounded by $[0, 2\pi]$ and it increases monotonically with $\delta\theta_t$ at each step t, the robot can visit only one location per θ_t.

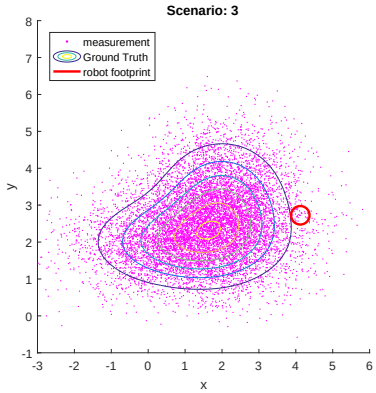
To access the performance of angle correction, we performed three experiments



(a) The GMM parameters for this radiation field are $\mu_1 = [1, 2], \mu_2 = [2, 5]$, $\pi_1 = \pi_2 = 1$, $\Sigma_1 = [2, 0; 0, .5], \Sigma_2 = [1, 0; 0, 1]$



(b) The GMM parameters for this radiation field are $\mu_1 = [1, 2], \mu_2 = [2, 2]$, $\pi_1 = \pi_2 = 1$, $\Sigma_1 = [2, 0; 0, .5], \Sigma_2 = [1, 0; 0, 1]$



(c) The GMM parameters for this radiation field are $\mu_1 = [1, 2], \mu_2 = [2, 3]$, $\pi_1 = \pi_2 = 1$, $\Sigma_1 = [2, 0; 0, .5], \Sigma_2 = [1, 0; 0, 1]$

Figure 5.2: **Radiation field:** the field is generated using GMM. The measurement of the field is represented by pink dots and the robot's initial position is denoted by red circle.

with three different scenarios. Fig. 5.3 shows the angle correction over time. While $\delta\theta_t$ is acquired at different points in time t, the initial $\delta\theta_0$ is the same for all algorithms. As we discussed earlier, the robot state denoted by $x_t = [\rho_t, \theta_t]^T$, and the origin of the robot's coordinate transformed to nearby random location x_0 where $z(x_0) > \beta$. Therefore, initially smaller ρ_t value resulting in coarse $\delta\theta_t$. As the time proceeds, the robot travels those locations where ρ_t is high value, resulting in the fine $\delta\theta_t$ by adaptive angle correction. Fig. 5.4 shows the estimated boundaries by different algorithms. This explains the direct consequence of the necessity of angle correction. It is also evident from the Fig 5.4 that the measurement noise makes it harder to obtain and maintain an accurate estimate because the error caused by it accumulated over time. However, it is not the fact that long time adaptive angle correction always yields an accurate estimate. We reported in Fig 5.3 that adaptive angle correction is not an issue when the measurement noise leads the robot in the wrong estimate. The more interesting observations reported in Fig 5.3. Note that unless the work in [22], the robot needs to explore the boundary of interest at most one time, resulting in the minimization of required exploration.

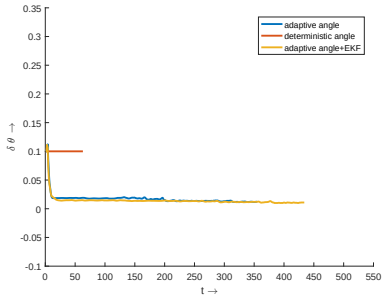
5.3.2 Estimation Phase

When the robot finishes its exploration phase, a smooth boundary is estimated along the traveled locations. we use polynomial curve fitting with order 4 to generate such a smooth boundary. The magenta lines in Fig 5.4 are the estimated boundaries for each exploration phase. In order to evaluate the estimation accuracy, we use the Hausdorff distance to measure how far estimated boundary space is from the Ground truth. Let $\mathbf{S}$ and $\mathbf{S}^*$ be two polynomial representations of the estimated boundary and the ground truth respectively. We then define their Hausdorff distance by

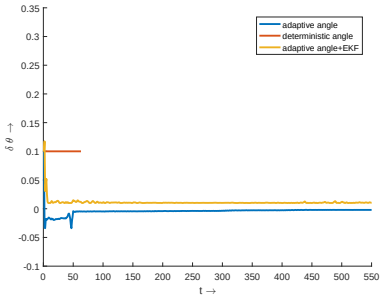
$$d_H(\mathbf{S}, \mathbf{S}^*) = \max \left\{ \sup_{S_x \in \mathbf{S}} \inf_{S_y \in \mathbf{S}^*} d(S_x, S_y) \, , \; \sup_{S_y \in \mathbf{S}^*} \inf_{S_x \in \mathbf{S}} d(S_x, S_y) \right\} \tag{5.20}$$

where d is the distance function, sup represents the supremum and inf the infimum.

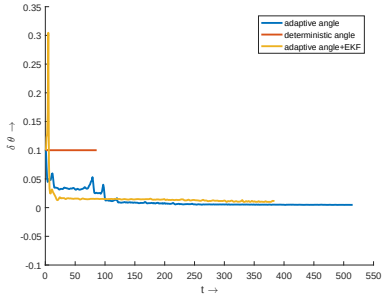
Existing works that estimate the environmental boundary require deterministic angle correction, prior estimation, revisiting some locations. In absence of prior estimation, we therefore propose adaptive angle correction with EKF to provide an accurate estimation while respecting the limited exploration budget. To compare the accuracy among the estimated boundaries, we performed 9 experiments. Table 5.1 summarizes the experiment results with the respect to Hausdorff distance.The estimated boundary



(a) In scenario 1, while the deterministic angle correction fails to track the nonconcave regions, adaptive angle with EKF efficiently correct the angle to track the nonconcave regions.



(b) In scenario 2. even though adaptive angle and adaptive angle with EKF exhibit almost similar performance, the angle correction ranges are different due to noise effect.



(c) In scenario 3, the adaptive angle with EKF minimize the redundant path caused by noise.

Figure 5.3: **Angle correction:** the performance of angle correction is demonstrated by varying 3 factors, namely, adaptive angle, deterministic angel and adaptive angle with EKF. The performance of the adaptive angle with EKF outperforms than others.

Table 5.1: Hausdorff distance for estimated boundaries

	Adaptive Angle	Deterministic Angle	Adaptive Angle+EKF
Scene 1	1.00 ± 0.07	2.36 ± 0.45	0.74 ± 0.05
Scene 2	0.30 ± 0.05	5.53 ± 0.09	0.23 ± 0.01
Scene 3	0.29 ± 0.2	0.87 ± 0.11	0.29 ± 0.00

with minimum d_H represents the better estimation accuracy. As it obvious from the table, the proposed boundary estimation shows the minimum value compared to others. This is likely due to the presence of the noise cancellation and an adaptive angular adjustment features. Despite those features, the d_H is relatively high and indicates poor accuracy.

5.4 Summary

Environmental boundary estimation is the process of bounding the region(s) where the measurement of all locations exceed a certain threshold value. In this chapter, we develop a framework for environmental boundary tracking and estimation in unknown environments. Dedicated sensors mounted on the vehicle considered to measure the boundary of interest in a point wise fashion. Focusing on the limited resources of Unmanned Aerial Vehicles (UAVs), it is important to track an unknown boundary in a fast manner. Therefore, we seek to a motion plan that leverages a single UAV to estimate the boundary of a given target area while minimizing the exploration cost. To do so, we improve the conventional PID controller based framework by integrating a noise canceling filter and a novel adaptive crossing angle correction scheme. The effectiveness of the proposed algorithm demonstrated in three different simulated environments. We also analyze the performance of framework subjects to proposed modifications.

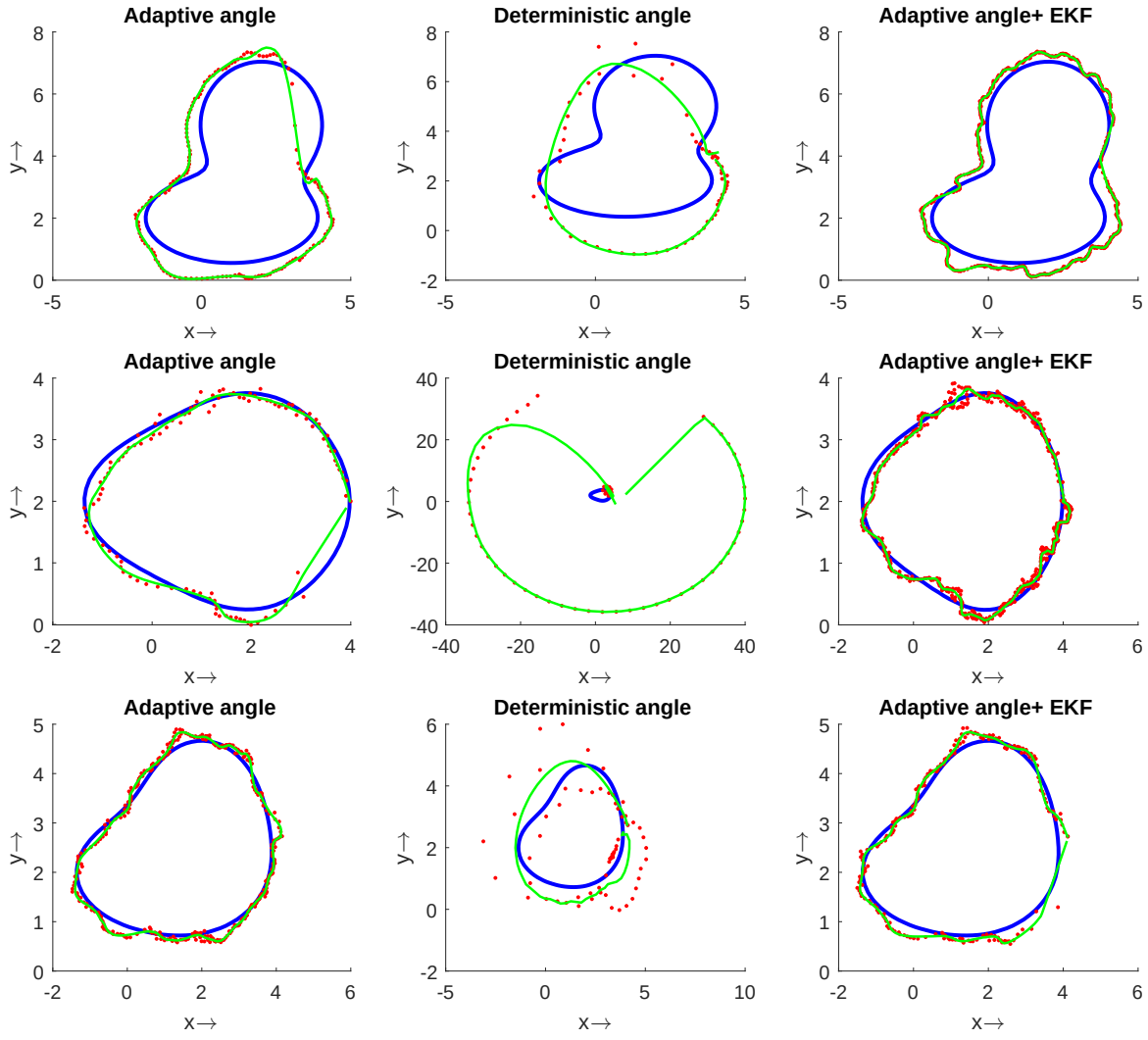


Figure 5.4: **Estimation Accuracy:** The robot's explorations represented by red dots. The blue contour line is the ground truth while the green contour line is estimated boundary. The adaptive angle correction with EKF always performed better among the others.

Chapter 6

Source Localization

Fig. 6.1 shows all the necessary steps of our proposed algorithm. Although our source position estimation is also based on a topographic map, the proposed approach differs significantly from [17]. First, from a given partial map, we find a set of interested measurement attributes (intensity values) coupled with the position information using our novel log gradient classifier. Starting with the initial positions, multiple contour lines are generated by tracking identical intensity values. Secondly, the ROI contour line is automatically chosen using contour shape analysis. Finally, instead of a single method -which cannot optimize all type of distributed sources, we propose a self-adaptive framework to select the appropriate method to localize the source positions in the most efficient manner.

6.1 Radiation Field Modeling

There are significant differences between geographic mapping and radiation distribution mapping [52]. In this work, we aim to include the radiation intensity and its distribution on top of a geographic map, based on the assumption that the UAV localization error is negligible. In our radiation mapping problem, we assume that a partial observation of the field is given, but the overall radiation distribution is unknown. Partial observation can be found, for instance, observing a UAV trajectory coupled with measurement attributes of the radiation field. It is necessary that the UAV trajectory connects an arbitrary intensity zone to the hotspot so that a rough estimation of the radiation field can be made initially. While the accurate radiation map can be obtained at the expense of exhaustively exploring the area, it is desirable to develop an efficient and effective mapping method considering the limited resources of UAVs. In this chapter, we categorize the field, so that the UAV does not need to visit all the terrains, but rather to explore only the ROI contour or the area bounded by the ROI contour to localize the

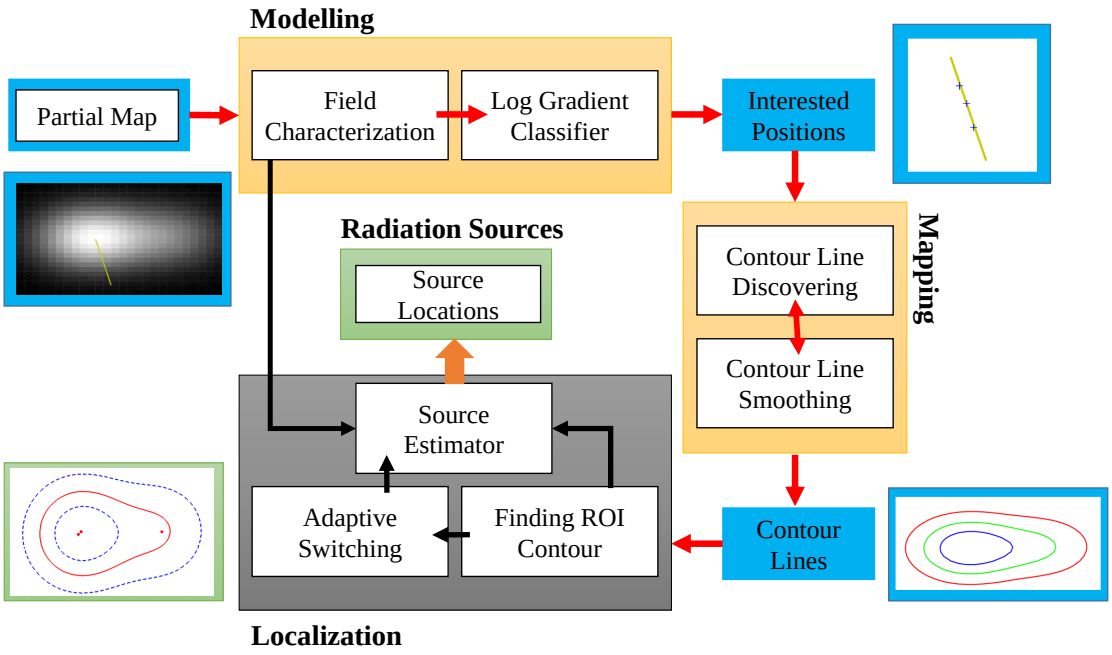


Figure 6.1: **System Overview:** A hotspot directed path coupled with measurement attributes is considered as a partial map for this system, it is denoted by the yellow line in the figure. The log-gradient classifier segmented to the partial map into a finite number of interested positions, denoted by (+) symbols. The topographic mapping processes generate the contour lines from the interested positions. The ROI contour is selected by the similarity analysis of the contour lines. Two methods are studied to balance the tradeoff between the rapid localization and the precise localization. Thus, the adaptive switching methodology selects the best method to pursue the rapid and precise localization objectives. The blue boxes are the output of each process. The yellow boxes, including red arrows, are the processes for ROI selection. The gray box and black arrows are the process of source localization.

sources.

In this section, we explain the procedure to characterize the radiation field using **GMM**. Next we explain how to incorporate prior knowledge. Using the log-gradient classifier, we segment the partial map into several interested positions.

6.1.1 Field Characterization

The intensity in the field could change gradually or abruptly depending on the source location. A region could have high intensity values because of the influence of multiple sources or the presence of strong sources nearby. Hence, it is not plausible to detect individual sources because of unknown diffusion information about each source. In order to predict their possible locations, we attempt to show their effect in the geographic map, adding the distribution of radioactive intensity on the map. In other words, the cumulative radiation effect of the sources is unlikely to be represented using a unimodal Gaussian model. For this reason we use **GMM** to characterize the radiation field. Let $x \in \mathbf{X}$ represent the location of the field and $z(x) \in \mathbf{Z}$ represent the corresponding measurement. The field property is characterized using GMM with M components such that

$$F_m(x; \alpha, \mu, \Sigma) := \sum_{b=1}^{M} \frac{\alpha_b \phi(x - \mu_b)}{{\Sigma_b}^2} \tag{6.1}$$

where $\phi(x) = \exp\left(-\frac{||x||^2}{2}\right)/(2\pi)$; $\mu_1, ..., \mu_M$ are the means; $\Sigma_1, ..., \Sigma_M$ are the variances and $\alpha_1, ..., \alpha_M$ are the mixing weights that describe the Gaussian components. The mixing weights are non-negative and add up to one. In order to generate the ground truth, we assume that each component has equal strength and the relative distance between each mean and measured location has influenced α. However, in our case α is equally divided by the number of sources (M), and the variance (Σ) is not important to localize sources. Therefore, we only need to estimate the mean (μ) of the sources using VB.

6.1.2 Log-gradient classifier (lgc)

The log-gradient classifier works like a rounding function for grouping the partial map. It converts the partial map into a finite number of interested positions based on the numerical relationship. Let $x_{i=0}$ be the robot initial position, $x_{i=h}$ be the hotspot location and i be the element index of the partial map, $i : \mathbb{R}^3 \rightarrow \mathbb{N}$. Also let the function $z(x_0, x_i)$ be the measurement attribute of the corresponding location x_i w.r.t. x_0 such that $z : \mathbb{R}^2 \rightarrow \mathbb{R}$. Let us draw a line as shown in Fig. 1(c) connecting the robot position.

The line also contains the measurement attributes based on the region of the colored radiation map. Therefore, the partial map is the set of a small section of the field including the corresponding measurement attribute. Our target is to group the partial map in an efficient way. First, we investigate how the measurement varies w.r.t. the robot position by taking gradient at a map index i given by

$$\nabla_i = \frac{z\left(x_0, x_i\right)}{d\left(x_0, x_i\right)} \tag{6.2}$$

where, $d(x_0, x_i)$ is the distance function w.r.t. the initial position of the robot. However, in order to group the different zones into the same layer, we rather focus on the power of gradient values given by

$$\log\left(\nabla_i\right) = \log\left(\frac{z\left(x_0, x_i\right)}{d\left(x_0, x_i\right)}\right) \tag{6.3}$$

The log-gradient operator classifies the partial map using Eqn. (6.3), which is dependent on the precision value, Λ, to get the number of classified regions $x_{\hat{\mathbf{j}}}$, where $\hat{\mathbf{j}} \in \mathbb{N}^m$ is the index set that contains a subset index of the partial map locations. In summary, the input of the log-gradient classifier (*lgc*) is a partial map which is a set of explored locations coupled with measurement attributes, $< x_{0:h}, z_{0:h} > \in \mathbb{R}^{n\times 3}$, and a user defined precision value Λ; while the output is the classified regions coupled with the measurement attribute $< x_{\hat{\mathbf{j}}}, z_{\hat{\mathbf{j}}} > \in \mathbb{R}^{m\times 3}$. Note that the dimension of the classified regions m is smaller than the dimension of the partial map n. The operation of lgc can be expressed as follows

$$\mathrm{lgc}\left(< x_{0:h}, z_{0:h} >, \Lambda\right) :=< x_{\hat{\mathbf{j}}}, z_{\hat{\mathbf{j}}} > \tag{6.4}$$

The sequence of interested positions starts from the hotspot location and terminates at the outward periphery such that $\hat{\mathbf{j}} = \{h : m\}$ in Eqn. 6.4. Intuitively, we can say that if Λ is a high value (say 8 digits after the decimal point), the classified regions are more in terms of dimensions, results in more number of interested positions shown in Fig. 7.2.

6.2 Topographic Mapping

The *lgc* provides the primary interested locations as well as their corresponding measurements, the whole contour line is discovered in the contour generation phase. We use the intensity information to track a contour line. It is known that intensity along the contour line is a constant value, the robot then discovers the contour line by mapping the measurement gradient into the geometric domain. However, it is beyond the scope of this chapter to summarize the state of the art in detector model; our overall

approach is concerned about the robotic exploration and is not restricted to a specific sensor model. Without loss of generality, we will restrict our discussion to the exploration strategy along with measurement uncertainties. Note that as defined by the *lgc*, we start contour discovering algorithm relative to the nearest distance from the hotspot so to adaptively stop contour generation phase.

6.2.1 Contour line discovering

Let us denote the robot position at time t as $\mathbf{x}_t = \{x_t, y_t\}$ in Cartesian coordinate. However, it is well known that drawing a contour line is relatively easier in Polar coordinate system, that motivates us to compute the robot position in Polar coordinate. The conversion from the Cartesian coordinate to the Polar coordinate system is simple and given by

$$\begin{aligned} r_t &= \sqrt{(x_t - x_0)^2 + (y_t - y_0)^2}, \\ \theta_t &= \arctan\left(\frac{y_t - y_0}{x_t - x_0}\right), \end{aligned} \tag{6.5}$$

where x_0 and y_0 are the initial positions. In order to draw a contour line, we transfer the reference point of the Polar coordinate to the hotspot location x_h. Let r be the radial distance from the reference point to the robot location. When the robot executes a control action, its next best location is given by a process model $\mathbf{x}_{t+1} = g(\mathbf{x}_t, r_t, \theta_t)$, where θ is the polar angle which range is 0 to 2π. By recursively updating the polar angle and the radial distance, a contour line over the robot positions can be discovered.

Fig. 6.2 (a) shows the geometric analysis of a contour line. Let, at each iteration step, us adjust the radial distance and the polar angle as follows

$$\begin{aligned} r_t &= r_{t-1} + \delta r, \\ \theta_t &= \theta_{t-1} + \delta\theta, \end{aligned} \tag{6.6}$$

where δr is the radial increment of r, where $\delta r \in \mathbb{R}_+$ and $\delta\theta$ is the angular increment of θ, where $\delta\theta \in \mathbb{R}_+$. Note that, if δr is 0 and $\delta\theta$ is $Constant$, the robot will discover a circular contour line. However, the goal of the contour discovering algorithm is to infer the robot position at each step based on knowledge of the observation of field strength (defined as intensity I).

Let c_r be the contour length in geometric domain while c_I be the contour length in intensity domain. The goal is to estimate the contour line in geometric domain by tracking the contour line in intensity domain. After each motion, the robot receives measurement attribute z_t of the field according to its current position x_t. The observation of the contour line in an intensity domain is given by a measurement model $I = h(z_t, z_\mu, z_w)$, where z_μ are the target intensity sets for the intensity tracking and z_w is some unknown

white noise added to model the measurement uncertainty. Starting from an arbitrary position on the contour line assigned by the lgc, the contour discovering algorithm repeatedly uses the measurement gradient information to map the geometric contour line by estimating an unknown scale λ such that

$$c_r = \lambda \cdot c_I, \tag{6.7}$$

where the length of c_r is given by

$$c_r = \int_0^{2\pi} r^2 + \left(\frac{\delta r}{\delta \theta}\right)^2 \delta\theta, \tag{6.8}$$

and the length of c_I is given by

$$c_I = \int_0^{2\pi} I^2 + \left(\frac{\delta I}{\delta \theta}\right)^2 \delta\theta. \tag{6.9}$$

Since the lgc initially assigned the values of r and I in Eqn. 6.8 and Eqn. 6.9, the main challenge is to compute the scale λ that maps the δI and δr. Next, we will show how to incorporate the measurement gradient information to accurately estimate the contour line in geometric domain.

6.2.2 Accurate Estimation

As it is obvious from Eqn. 6.6 that at each iteration step the robot has to infer the radial distance of the next location, the measurement uncertainties as well as the prediction uncertainties may lead the robot along the inaccurate contour line. Therefore, in this subsection, we introduce a recursive Bayesian filter to cope with the above mentioned uncertainties.

The contour line over the geometric domain can be discovered by discretizing the polar angle θ_t into a constant increment $\delta\theta \in \mathbb{R}_+$. Given the current polar angle θ_t, the robot estimates the radial increment $\delta r_t|\theta_t$ at each iteration step. After that, it computes the intensity gradient $\delta I_t|\theta_t \in \mathbb{R}$ as follows

$$\delta I_t|\theta_t = z_t - z_\mu + z_w \tag{6.10}$$

where z_μ is the target intensity, z_t is the current measurement attribute and z_w is Gaussian white noise. Note that the intensity gradient could be a positive or a negative value depending on the sensing location. In order to apply this gradient information in geometric domain, we use a recursive Bayesian filter to compute the optimal scale λ given

by

$$\lambda = \delta r_t|\theta_t \cdot \delta I_t|\theta_t \cdot \left(\delta r_{(t-1)}|\theta_t\right)^{-1} \tag{6.11}$$

Since the recursive Bayesian filter summarizes the current estimation $\delta r_t|\theta_t \cdot \delta I_t|\theta_t$ with the respect to the immediate past $\delta r_{(t-1)}|\theta_t$, as more observations are made, the scale will begin to converge to the true value. However, given a polar angle θ_t, when the estimated scale λ is a large value, the robot has to travel a long radial distance r, which can cause a huge prediction uncertainties. If we do not minimize that uncertainties, it will be propagated throughout the Bayesian filter into the future estimation, resulting in poor estimation accuracy.

Here we propose to optimize the polar angle increments to compensate for the discrepancy between the desired polar angle θ_t and the radial distance r as follows

$$\delta\theta_t|r_t = atan\left(\frac{\delta\theta_t^2}{r_t^2}\right) \tag{6.12}$$

In this case a large radial increment is discretized by reducing the polar angle given by

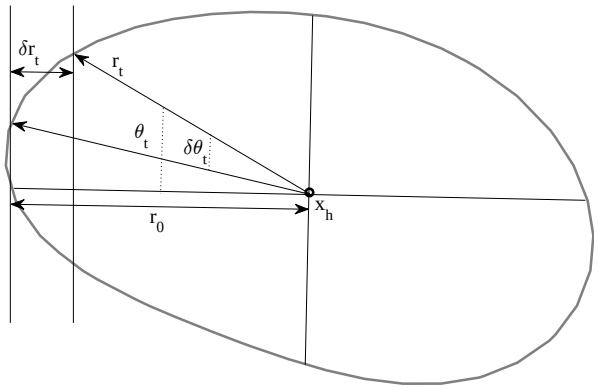
$$\theta_t = \theta_t - \delta\theta_t|r_t \tag{6.13}$$

Algorithm 5 summarizes the overall estimation method. Note that each iteration step the predicted polar angle increment is a constant value, but the effective polar angle update is consistent with the optimal scale. Fig. 6.2 (b) represents the exploration strategy to discover an unknown contour line in the geometric domain.

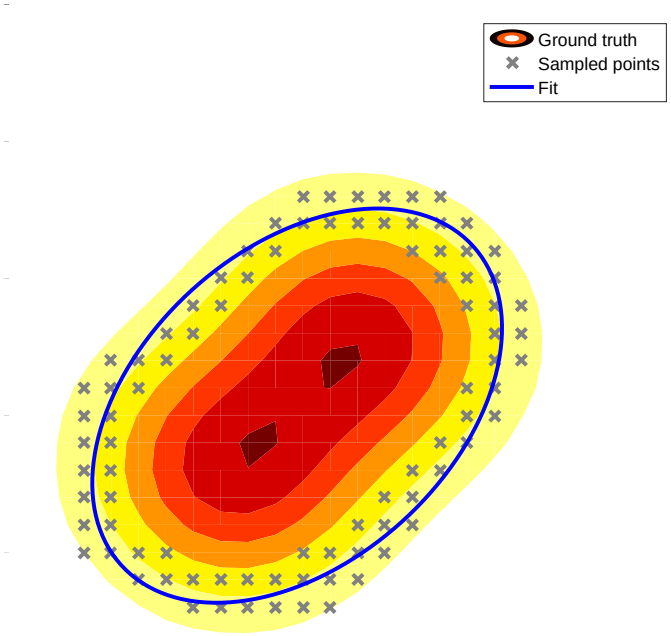
6.2.3 Finding the ROI contour

A topographic map may contain multiple contour lines depending on the designed parameter (Λ). However, all the contour lines are not important to explain the characteristic property of distribution. Obviously, contours near to the hotspot region are very important, since they help us visualize anisotropic, dynamic changes in the intensity of the radiation field. As the contour line goes outwards from the hotspot, the shape tends to be quite similar to each other. Therefore, we can analyze the contours shapes whereby the robot can terminate the exploration.

The previous contour line discovering process was designed to discover all the contour lines. Based on that process, we can find $\hat{\mathbf{c}} = \{c1, c2, .., cl\}$, which is the set of contour lines. Note that c_r, c_I are the length of each contour line in the geometric domain and the intensity domain respectively, whereas $\hat{\mathbf{c}} \in \mathbb{R}^l$ is the index set of all contours and l is the dimension of that set. We consider the global Cartesian coordinate system in order to analyze the degree of similarity between neighboring contour lines.



(a) Analysis of a contour line geometry.



(b) Contour line exploration

Figure 6.2: Contour discover: A contour is discovered by recursively updating the radial distance and the polar angle using a Bayesian filter.

Algorithm 5 Contour discovery

Require: initial r, initial θ, initial δr, constant $\delta\theta$
Ensure: Contour line in geometric domain

1: **Predict radial distance**
$r_{(t|t-1)} = r_{(t-1|t-1)} + \delta r_t|\theta_t$
2: **Predict polar angle**
$\theta_t = \theta_{t-1} + \delta\theta$
3: **Compute measurement residual**
$\delta I_t|\theta_t = z_t - z_\mu + z_w$
4: **Compute optimal scale**
$\lambda = \delta r_t|\theta_t \cdot \delta I_t|\theta_t \cdot \left(\delta r_{(t-1)}|\theta_t\right)^{-1}$
5: **Update radial increment**
$\delta r_t = \lambda \cdot \delta r_t$
6: **Update radial distance**
$r_t = r_t + \delta r_t$
7: **Update polar angle increment**
$\delta\theta_t|r_t = atan(\delta\theta_t^2/r_t^2)$
8: **Update polar angle**
$\theta_t = \theta_t - \delta\theta_t|r_t$

For this we introduce the elements σ_x and σ_y, at each exploration in the contour discovering process to compute the relative changes of initially assigned radius, r_0, to the radius, r_t, at current exploration step given by

$$\begin{aligned} \sigma_x &= \{(r_t - r_0)\cos(\delta\theta)\}, \\ \sigma_y &= \{(r_t - r_0)\sin(\delta\theta)\}. \end{aligned} \tag{6.14}$$

It is obvious from the above equation that $\hat{\boldsymbol{\sigma_x}} = \left\{\bigcup_t {\sigma_x}^t\right\}$ and $\hat{\boldsymbol{\sigma_y}} = \left\{\bigcup_t {\sigma_y}^t\right\}$ represent the change in the radius w.r.t. the global Cartesian x-axis and y-axis respectively. Next, we will analyze the similarity between two neighboring contour lines by defining a function given by $\Gamma : \mathbb{R}^{l\times 2} \rightarrow \mathbb{R}$. We compute a score for each contour line w.r.t. the neighboring contour line closer to the hotspot using the following equation such that

$$\Gamma^* = \tan^{-1}\frac{E\{\hat{\boldsymbol{\sigma_x}}\}^2}{E\left\{\hat{\boldsymbol{\sigma_y}}\right\}^2} - \Gamma, \tag{6.15}$$

where Γ^* is the current contour score and Γ is the neighboring contour score. When Γ^* reaches a predefined tolerance limit, adding a new contour would be redundant. Therefore, the robot can stop its exploration and narrow down the **ROI** to the previous contour such that

$$ROIC = \arg\min_l \left\{\max_{\Gamma^*\in\Gamma^l}\left\{\Gamma^l\left(cl, c(l-1)\right)\right\}\right\}. \tag{6.16}$$

where, $ROIC$ is the region of interested contour index.

6.3 Radiation Sources Localization

In this section, we briefly explain two different approaches to the localization of the radiation sources, namely HT and VB inference. The HT approach is reasonably accurate if we deal with the clustered sources only. It can localize the source position by exploring only to the ROI contour line. When the source distribution is somewhat uniformly scattered or biased, a potentially more accurate estimation than the HT can be obtained by the VB inference. The particular VB approach used here is based on the importance sampling which involves drawing samples from the ROI. In the context of ROI, a dense sampling can be performed within the bounded region, so that the estimation of the VB can converge to the true source positions. Note that the main VB algorithm takes account of measurement uncertainties while estimating the source positions. Therefore, we will skip the additional explanation of the measurement uncertainties for the VB algorithm. On the other hand, since the input of HT is a contour line, the uncertainties caused by the measurement instabilities are already discussed in the earlier section.

6.3.1 HT based source localization

The Hough Transform (HT) is a very useful tool to solve the computer vision and the image processing problems. Although HT is typically used to detect lines, it has been applied to the radiation sources detection problem in [17]. Since only contour lines are required to localize sources, it should be straightforward to use the conventional HT in the rapid source localization mission. In order to implement HT to localize the sources, first, the detected contour lines are converted to a binary image. Secondly, the binary image is downsampled from one half to one quarter the original resolution to reduce the computational burden. Thirdly, the Hough circles are chosen by deterministically defining the maximum and minimum lengths of the radius parameters. Finally, the transform process can determine the center of each circular distribution.

However, for the HT has several major disadvantages, especially when applied to unknown radiation distributions. First, the determination process of the Hough circles is sensitive to the radius parameter. Without a good radius parameter, HT cannot accurately estimate the source positions. Furthermore, it is observed that HT can perform well when sources are clustered within a comparatively small area and their effects are approximated by a Gaussian distribution. It can therefore be concluded that, the applications of the HT in the source localization process are likely to be limited and source distribution type-specific.

A significant advantage of the HT over the VB is that it does not require information regarding the whereabouts of ROIs. We therefore use the HT to optimize exploration for the clustered sources, where the ROI contour localizes the sources with reasonable accuracy. The pseudo-code for the HT algorithm is given in Algorithm 6. In our simulation the convenient radial upper bound is the size of the image, while the lower bound is one ninth of the image size. The matlab function we use here is $houghcircles$ that detects the center of the radiation sources.

Algorithm 6 Hough Transform

Require: ROI contour
Ensure: Sources
1: **Convert the contour to a binary image**
$cImage = cnt2bin(contour)$
2: **Determine the parameters**
$maxRadius = size(cImage)$
$minRadius = maxRadius/9$
3: **Localize the sources**
$centers = houghcircles(cImage, maxRadius, minRadius)$

6.3.2 VB inference based source localization

The exploration goal of the robot in this subsection is to gather the measurement attribute for a set of sensing locations which is bounded by the ROI contour line. Estimating radiation sources with given a set of measurements is commonly referred to as inverse problem. The inverse problem is difficult by the fact that different regions could have the same intensity value. However, in our proposed framework, the ROI contour line can provide an additional information of the measurement distribution in geometric domain. Therefore, we here contribute a probabilistic kernel function, so that the distribution of sensing locations along with their corresponding measurement attributes can jointly compute the maximum likelihood for the VB algorithm, that results in improved estimation accuracy.

Let $\mathcal{P}$ be the exploration path that traverses each of the sensing location at most one time. With slightly abusing the notation, assume that a location of the path $x_s \in \mathcal{P}$ pairs with the measurement attribute of that location z_s generates a sample point $< x_s, z_s >$. After traveling the path $\mathcal{P}$, the robot finds all the sample points bounded by the ROI contour line, which is then used to compute a probabilistic kernel function given by

$$k(\forall z_s) = \frac{1}{2 + \exp(\forall z_s)}. \tag{6.17}$$

Although we have gathered the measurement attribute, the actual sources are hidden

variables. In order to estimate the source positions, firstly, we characterize the ROI area as an unknown GMM. Secondly, based on the measurement attribute, the parameters of the GMM are obtained by computing maximum likelihood. The goal of the kernel function is to combine the geometric shape information and the measurement attributes as follows

$$p(\forall x_s|\forall z_s) = \mathcal{N}(\forall x_s, \mu\left(x_s\right)) - k(\forall z_s), \tag{6.18}$$

where $\forall x_s$ is a column vector that contains all the sensing locations information and $\forall z_s$ is also the column vector of corresponding measurements, thus, the normal distribution $\mathcal{N}(\forall x_s, \mu\left(x_s\right))$, the probabilistic kernel $k(\forall z_s)$ are also a column vectors respectively. Finally, the optimal number of components for Bayesian GMM can be obtained iteratively using a variational EM algorithm [93], which is achieved through partially performing an E-step and observing the maximization of E-step and M-step using the same function $F[q, \pi]$ such that

$$F\left[q, \pi\right] = \sum_{x \in \forall x} \int q\left(\forall z, \mu, \mathrm{T}\right) \log \frac{p\left(\forall x, \forall z, \mu, T; \pi\right)}{q\left(\forall z, \mu, \mathrm{T}\right)} \partial\mu \partial T, \tag{6.19}$$

where the parameters (μ, T, π) are the mean (center) of the sources, the precision matrices and the mixture weights respectively. q is the arbitrary distribution that approximates the posterior distribution defined by

$$p\left(\forall x, \forall z, \mu, T; \pi\right) \stackrel{def}{=} p\left(\forall x|\forall z, \mu, T|\forall x; \pi\right). \tag{6.20}$$

From Eqn. (6.18), we can see that uniform sample points inside the **ROI** area are explicitly biased toward the significant measurement attribute, which results in conversion of cluster samples. Therefore, the VB can easily estimate the optimal number of sources and their corresponding locations. The detailed formulas for computing the parameters can be found in [93]. In summary, at each iteration, the VB performs two following steps:

- **Variational E-Step:** Evaluate $q^* = \arg\max_q F\left[q, \pi\right]$
- **Variational M-Step:** Find $\pi^* = \arg\max_\pi F\left[q^*, \pi\right]$

A notable property of this model is that when maximizing F, the prior distribution of μ and T penalizes the overlapping components, therefore the redundant sources whose effect are negligible to the distribution, are eliminated. Furthermore, it is sufficient to find the mean μ components to estimate the source position. We use an open source MATLAB function [94] to implement VB.

Algorithm 7 Variational Bayesian

Require: ROI contour, exploration path $\mathcal{P}$
Ensure: The number of sources and their corresponding positions
1: Generate the sample points by traveling $\mathcal{P}$.
2: Compute the probabilistic kernel function using the Eqn. (6.17).
3: Compute the maximum likelihood function using the Eqn. (6.18).
4: Estimate the number of sources and their corresponding positions using the Eqn. (6.19).

6.3.3 Adaptive switching strategy

The HT and VB methods described in earlier sections are now selectively used. We propose a switching method supported by the ROI selection scheme presented in the previous section, allowing the method to rapidly converge to a solution despite the lack of prior knowledge of the radiation field.

Similar to the ROI selection process, the variance slope of each contour lines can be used to estimate the distribution of source positions. Note that the sources are different than the measurement distribution. According to our findings, the variance slope as described in the previous section exhibits the following three characteristic properties such that

1) **Increasing order of the slope gradient :** In this category, the variance slope decreases from the outer periphery to the hotspot periphery. We observed that biased sources exhibit this type of behavior, because the inner contour line of biased sources is almost circular and outer contour lines are elongated along a specific direction. Thus, the variance slope converges from an elliptical and irregular circular shape to a (nearly) round shape.

2) **Decreasing order of the slope gradient :** Unlike the previous definition, the contour shape of a scattered sources is propagated from the nearly round to elliptical and irregular circular shapes while approaching to the inner contour. Therefore, the variance slope exhibits a decreasing property while stepping toward inner contour lines.

3) **Constant order of the slope gradient :** When sources are positioned close to each other, all the contour lines detected by the exploration, basically turn out to be uniform circular shapes. Thus, the variance slope remains at a nearly constant level. We found that this is the case for the clustered sources.

An optimized active exploration performance, as described in the earlier section, can be achieved by selecting an to appropriate method. In detail, the HT can be applied only

for the constant order of the gradient slope property. On the other hand, the VB can be used to tackle the increasing and decreasing order of the gradient slope properties. Choosing an appropriate method by this strategy alleviates the need for superfluous exploration. It is observed that without the proposed selection method can localize the sources with improved accuracy. However, the challenging part is fast and accurate analysis of the ROI contour.

6.4 Simulation Result

We have performed an extensive simulation validation of our algorithm in different settings of the sources. Our first experiment focuses on reducing ROI in the radiation field depending on measurement distribution. Next, we extend the experimental settings to evaluate the source localization strategy. Finally, we analyze the effect of the ROI selection to the source localization, and also show that the proposed adaptive method optimizes the tradeoff between exploration and localization accuracy. However, the partial map given to this system does not depend on specific initial positions. It just contains the rough idea of the intensity distribution from lower to higher zones. In summary, we have shown that the ROI selection is very important since it can reduce the traversed path and enhance the estimation accuracy. Furthermore, we have also shown the under which condition we can extremely reduce the traversed path.

6.4.1 Reducing ROI

The partial map of the environment contains the UAV trajectory and corresponding measurement up to the hotspot location. The *log-gradient classifier (lgc)* classifies the trajectory depending on the measurement change. The main advantage of *lgc* is that it automatically segments the trajectory depending on the numerical properties of the measurement, resulting in a finite number of groups. Each group contains the starting position and the corresponding measurement value. It is then further explored to determine the whole line through the contour discovering process. We perform three experiments in determining ROI contour where sources are distributed in such form as scattered, clustered and biased respectively.

Fig. 6.3 represents our experimental situations, where the contour lines are drawn by mapping the intensity changes into the geographic domain. The background gray colored map is the distribution of the measurement, while the yellow line represents the given trajectory of UAV, which is also the partial map that fed to *lgc*.

Although *lgc* segmented the field into a finite number of groups, the similarity analysis of contour shape allows us to reduce the contour numbers further more. The simi-

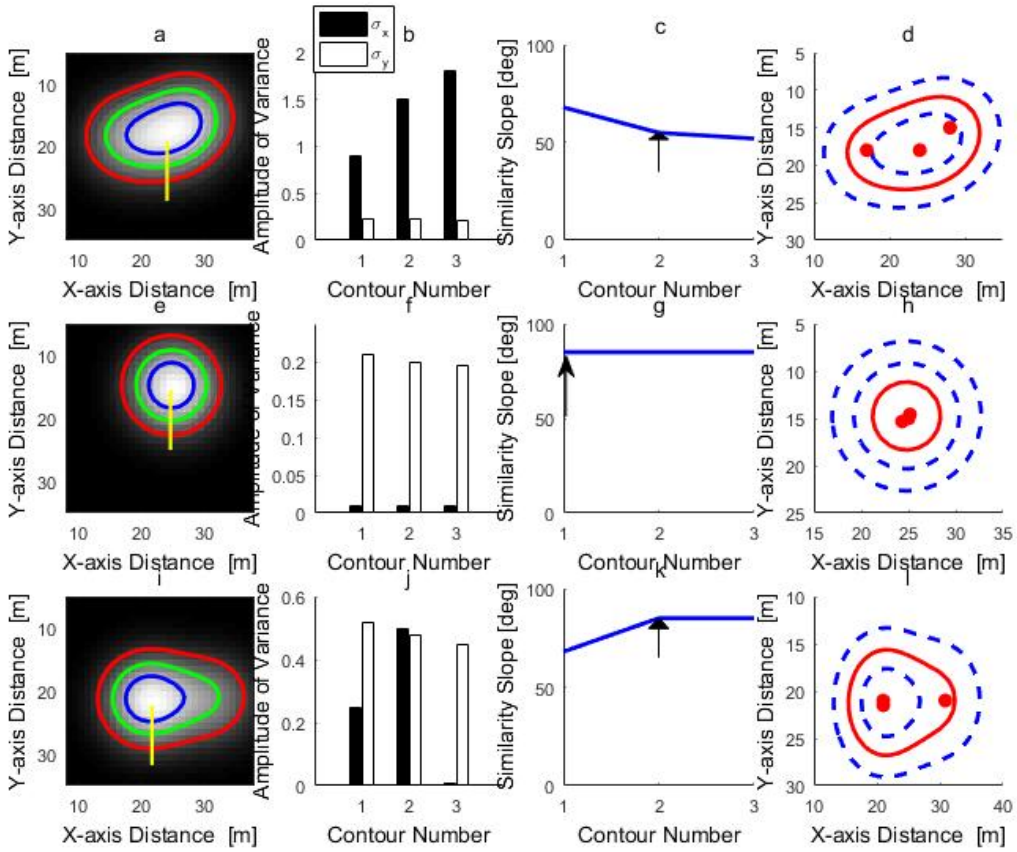


Figure 6.3: **Finding ROI contour:** The evaluation of the ROI contour computed by similarity analysis. Three different experiments are conducted namely scattered sources (a-d), clustered sources (e-h) and biased sources (i-l). The blue, green, red contours in (a,e,i) are labeled as (1,2,3) in (b, c, f, g, j, k). The variance of each contour is computed over circular path while the similarity slope between two consecutive contours is computed using Eqn. (6.16). The arrow in (c, g, k) indicates the starting position of similar contours. Finally the red contour line shown in (d, h, l) represents the ROI contour where the red dots are the actual sources.

larity slope varies depending on the distribution. As can be seen in Fig. 6.3 (c), (g), (k), the similarity slope between two consecutive contours reaches to a saturation level after a certain period. When the slope gets saturated, we can discard the current contour and fix our ROI onto the previous one, which explains why the ROI contours in Fig. 6.3 (d), (h), (l), are $2, 1, 2$ respectively.

6.4.2 Source Estimation

In this scenario, we have extended our experiments to source localization. After determining the ROI contour, we consider how the sources are localized. Fig. 6.4 shows the overall procedure of the algorithm, where the partial map in Fig. 6.3 (i) is discretized

using the *lgc* and a finite number of contours are drawn using the topographic mapping process. Among the traversed contours, the ROI contour is selected for further exploration. Samples are taken uniformly from the area bounded by the contour. The red circles in Fig. 6.4 (b), (e), (h), are the uniform sample locations. It can be seen

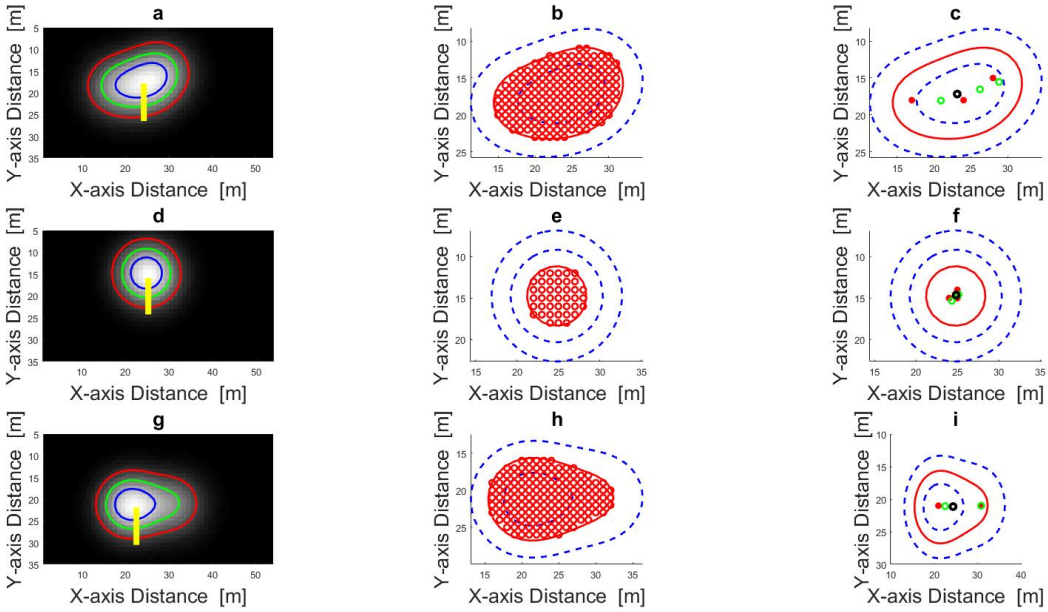


Figure 6.4: **Source Localization:** A radiation field is classified into a finite number of contour lines in **(a, d, g)** using log gradient classifier. Contour generation process is automatically terminated depending on similarity in shape analysis and uniform samples are taken inside the ROI contour in **(b, e, h)**. In **(c, f, i)** red dots are the actual sources, black circles are the estimated sources by Hough transform and green circles are the estimated sources by proposed algorithm.

Since the number of estimated sources is not equal to the actual sources, the performance of algorithm is measured by computing the distance to the nearest estimated source. Table 6.1 shows the difference between the VB and the HT. NDS1, NDS2, NDS3 are the Euclidean distance between the nearest estimated source and the actual sources, respectively. In Fig. 6.4 (c), (f), (i), the red dots are the actual sources while the black circles and the green circles are the estimated sources using the HT and the VB algorithm, respectively.

The performance achieved by VB is outstanding and very close to the original source location. It takes at most 264 iterations to converge to the resulting state. This improvement is achieved with a gathering of real measurement data inside the ROI contour. There are several reasons that the estimated sources do not exactly converge to the true state. This could happen because of the linearization error and the inverse problem [18]. Despite the variation in the true source positions, the worst case estimated error for the VB is $4.490m$ while the maximum estimated error for the HT is $10.837m$.

Table 6.1: Sources estimation

Src. type	Method	No. Src. (ground truth)	NDS1	NDS2	NDS3
Scatter	VB	3 (3)	4.490	2.618	1.942
	HT	1 (3)	4.490	2.618	7.758
Cluster	VB	2 (3)	0.778	1.399	1.604
	HT	1 (3)	0.778	1.408	1.707
Biased	VB	2 (3)	2.570	0.998	2.502
	HT	1 (3)	10.837	0.998	10.687

6.4.3 The Effect of ROI selection in localization

The selection of ROI contour is particularly important because the superfluous sources are eliminated as the method converges to a solution, thereby leading to an accurate localization of the sources. To visualize the effect of ROI selection, we have repeated previous experiments, and for each of them, sequentially selected all the contour lines, and estimated the source positions using the VB. In this setup, four contour lines are

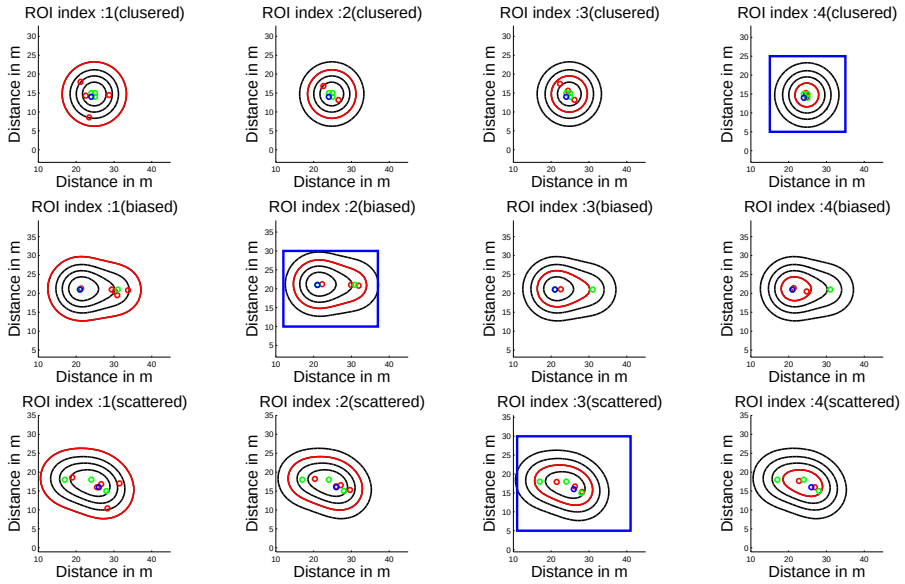


Figure 6.5: **The effect of ROI selection :** In this simulation, we compare the estimation accuracy w.r.t. the ROI selection. The ROI contour selected by the proposed strategy is denoted in the subfigure using the blue rectangular box. The source estimation simulations are carried out by the selection of a contour line starting off the outer periphery of the distribution. The red, blue and green circles are the estimated sources by the VB, estimated sources by the HT and the ground truth positions. It is observed that the ROI selection not only reduces the exploration space but also enhances the estimation accuracy.

assigned by the proposed classifier. The index of the contour line is counted from the

outer periphery. The effect of ROI can be seen in Fig. 4, which presents several results with the selection of the different ROI contours. From the analysis of Fig. 6.4, one can easily infer that the contour indexed 4, 2 and 3 are the ROI contours if we use the proposed strategy. In order to analyze the effect of the ROI selection, we compute the estimation error similar to the algorithm 8 for each contour index and plot them in Fig. 6.6. The estimation errors are also shown for the clustered, biased and scattered sources. Comparing to the ground truth denoted in the same Fig. 6.5, it is obvious from Fig. 6.6 that most estimation errors converge to a minimum level with the proposed ROI selection. Even though the proposed algorithm failed to show a minimum estimation error for the scattered sources as in Fig. 6.5, we can see that all the sources are bounded by the ROI contour line and the selected ROI contour is very close to the smallest loop, numerically, less than $2m$ away from the actual sources. This performance is achieved by a tight lower bound on the ROI area. The measurement likelihood is then generated with high-density sampling in the ROI area, as it is well known that the performance of VB excels with the increment of sampling density [95]. However, if ROI contour fails to enclose all the sources, the performance of VB deteriorates due to inadequate samples.

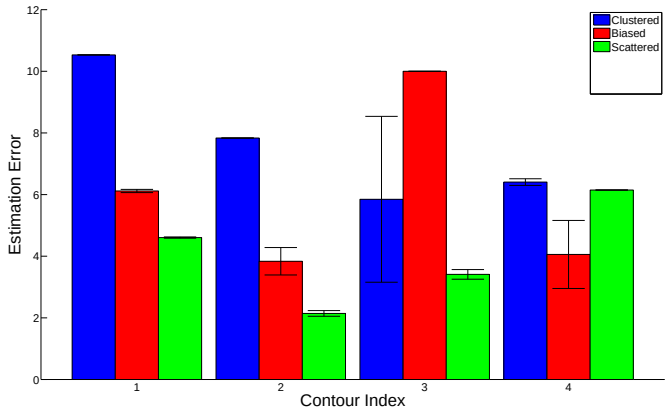


Figure 6.6: **Estimation error:** The estimated errors are using Algorithm 8. The results are computed with the mean over 100 simulations where the error bar represents the variation. The blue, red and green bars represent the estimation error for the clustered, biased and scattered sources. The minimum errors for the clustered, biased and scattered sources are found for the contour indices 4, 2, 3 respectively while the selected ROI contour indices using the proposed strategies are 4, 2, 2 respectively. Even though the proposed strategies does not find the optimal solution for the scattered sources, the estimation error is very close to the minimum error and bounded by the $2m$ distance.

On the other hand, the estimation of the HT depends on the geometric shape of the contour lines. Fig. 6.7 shows how the estimation of the HT could be changed depending on the geometric shape. However, in the case of round shapes, the HT has always found the same center of the distribution even though the ROI contour is different. Thus, the

ROI selection does not have a significant impact on the source localization, but only to prohibit the UAV from performing additional contour discovering processes.

6.4.4 Performance of the adaptive framework

Comparing the localization accuracy and considering the exploration constraints, the proposed adaptive method is a very efficient yet accurate solution. In general, we have found that there is a way out to optimize the localization process if the ROI contour can be accurately selected and analyzed. For comparing the performance of the adaptive method, we applied a slightly different metric: we looked at the estimation error and the length of traveled path which is required to perform each of the algorithms. Fig 6.8 shows the error convergence properties of each method. Since the estimated sources are different w.r.t. the ground truth sources, we then compute the average estimation error w.r.t. the nearest estimated sources similar to Algorithm 8. The simulations were performed into two phases. In the first phase, all the simulations were conducted independently without considering the ROI selection method. In order to compute the estimation error, we combine 100 simulation results and plot the mean estimation error with variance. Even though the ROI selection does not have any influence to estimate the sources in the first case, it is obvious from the Fig. 6.8 that the VB outperforms the HT except for the clustered sources.

Table 6.2 demonstrates the efficiency of the proposed framework. It is interesting to note that the sensitivity of the source localization manifests in the ROI selection criteria. Looking at Table 6.2, one can see that the estimation errors computed by the VB along with the ROI and the HT along with ROI are very close only for the clustered sources, numerically $0.95m$ and $1.15m$. In that situation, a rapid solution without any additional exploration in the ROI can then be generated using the HT, resulting in $19.77m$ path to travel instead of $30.84m$ path.

However, the significance of the ROI selection can be verified by the Fig. 6.8 (b). While the maximum and minimum mean estimation error without the ROI selection method were around $7m$ and $3m$ respectively, in the second case (with the proposed ROI selection method), the maximum and minimum errors converged to $2.25m$ and $0.85m$ respectively. It is observed from the table 6.2 that regardless of the specific ROI, the VB always outperformed the HT. However, in the first case, the estimation error of the VB is more sensitive for the biased sources. As observed in Fig. 6.6, the wrong ROI selection caused a large estimation error. Thus, the better results can be achieved only with the appropriate ROI selection. It is also obvious from the table 6.2 that the number of samples points without a ROI selection method cannot improve the estimation accuracy. As a result, the path required for the VB algorithm is usually longer without the ROI

Table 6.2: Exploration efficiency

Parameters	**Clustered**		**Biased**		**Scattered**	
	Path Len. (m)	Estimation Error (m)	Path Len. (m)	Estimation Error (m)	Path Len. (m)	Estimation Error (m)
a path without ROI + VB	226.92	5.5	330.85	3.11	352.55	3.25
a path without ROI +HT	53.44	5.95	66.78	3.25	68.41	7.02
a path with ROI + HT	19.77	1.15	51.16	2.25	41.01	2.05
a path with ROI + VB	30.84	0.95	189.00	1.25	122.18	1.75

selection than the path with the ROI selection. Despite the more sampling points by the longer path, the estimation accuracy is always better in the case of a path with the ROI selection, shown in the table 6.2.

Algorithm 8 Estimation error computation

Require: $source, estimation$
Ensure: $averageError$
$mse \leftarrow \{\}$
$averageError \leftarrow 0$
for $i = 0$ **to** $size(source)$ **do**
  for $j = 0$ **to** $size(estimation)$ **do**
    $x_s \leftarrow source(i, 1)$
    $x_e \leftarrow estimation(j, 1)$
    $y_s \leftarrow source(i, 2)$
    $y_e \leftarrow estimation(j, 2)$
    $mse \leftarrow mse \cup sqrt((x_s - x_e)^2 + (y_s - y_e)^2)$
  end for
  $averageError \leftarrow averageError + min(mse)$;
end for
$averageError \leftarrow averageError/size(source)$;

Since HT generates optimal results for clustered sources, we can then extend its applications to a collection of isolated sources. However, the localization of isolated sources is beyond the scope of this chapter. For a collection of a point sources, if one begins with the VB method which is explained in this chapter, then the VB can converge to a solution but with poor estimation. The reason why the performance of VB is poor is that our proposed method is designed focusing on the single hotspot with multiple sources. In the case of isolated sources, there could have been multiple hotspots and the measurement attributes are not evenly distributed throughout the target area. These results support two conclusions. First, the estimation of the VB always provides the better solution than the HT at the expense of additional exploration. Second, HT can drastically reduce the exploration expense, but the desired results can be obtained only for the clustered sources.

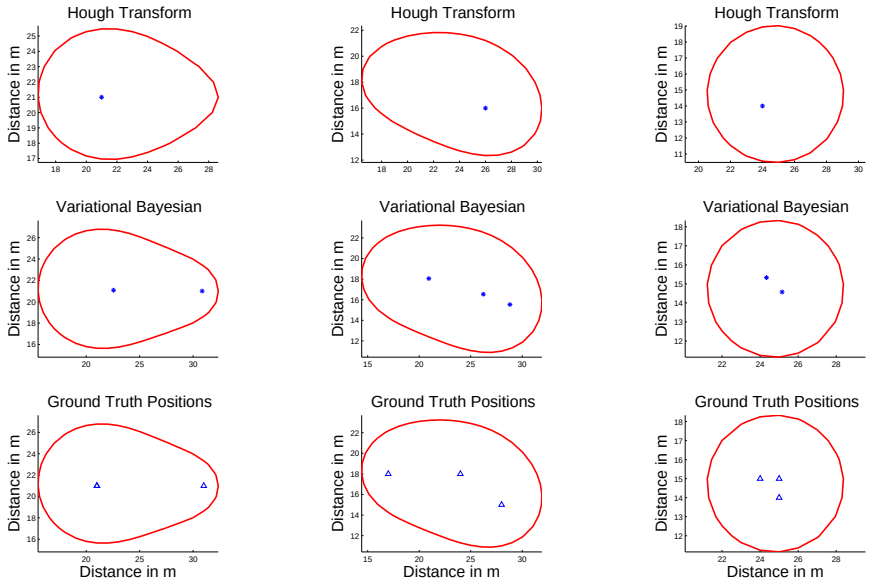
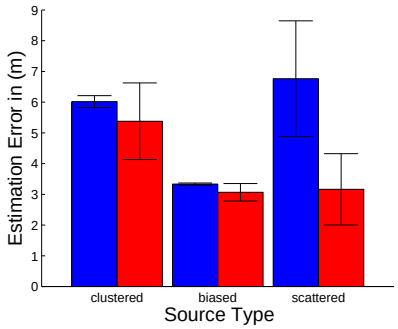


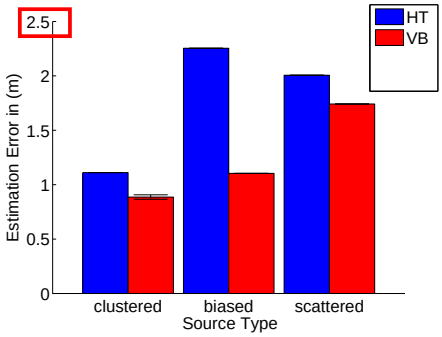
Figure 6.7: **Performance analysis :** Source localization simulations are carried out by the HT and the VB independently. The simulations are conducted in three types of spatially distributed sources, namely, biased, scattered and clustered. It is observed that the localization accuracy of HT is better only for the clustered sources. In the case of biased and scattered sources, the VB leads the solution to the close proximity of the ground truth positions. The ground truths are shown in the last row with the triangular shape.

6.5 Summary

In this chapter, we propose an efficient approach to the multiple source localization and contour mapping problem of radiation fields using Unmanned Aerial Vehicles (UAVs). A typical radiation field originating from a single hotspot can be generated by three spatial distributions of sources; scattered, clustered and biased. Of these, the clustered sources are relatively easy to localize, because the sources are located in a close proximity to the center of distribution. In other cases, it is not very straightforward, because, when multiple radiating sources generate a hotspot in a cumulative manner, sources do not coincide with the hotspot position. Regardless of our knowledge about the hotspot position, we attempt to solve the multiple radiation localization problem in two steps: the Region Of Interest (ROI) selection and the source localization. Existing algorithms eventually explore whole area, causing the problem of excessive use of UAV resources. We therefore propose a framework to reduce ROI in a radiation field that not only optimizes the resources but also increases the localization accuracy. For the source localization process, two different methods are employed interchangeably. Those methods are called the Hough Transform and the Variational Bayesian, adaptively selected with a switching



(a) without the ROI selection



(b) with the proposed ROI selection

Figure 6.8: **The analysis of error convergence :** The performance of the HT and VB is evaluated w.r.t the nearest estimated source location. The estimation error is computed over 100 iterations using algorithm 1 with (a) all possible combinations of the ROI selection (b) with the proposed ROI selection. The red bar is the estimation error of the VB and the blue bar is of the HT. It is obvious that the VB outperforms the HT in terms of estimation accuracy. It note worthy that the outstanding error convergence can be obtained only with the proposed ROI selection method.

technique and the overall performance is evaluated by balancing between the localization accuracy and the required exploration. In favor of the optimization, the prediction model defines the type of sources in a way that the adaptive switching methodology can converge to an optimal solution by selecting an appropriate method. Thus, the proposed framework enables the UAV to accurately localize the radiation sources in a fast manner. In order to verify the validity and the performance of the proposed strategies, we performed extensive numerical experiments with different numbers of sources and their positions. Our empirical results clearly show that the proposed approach outperforms existing individual approaches.

Chapter 7

ROIs determination

Our work presents the first opportunistic and iterative environmental boundary estimation for area coverage problem. The methodology is evaluated using two different strategies (namely, boundary estimation and coverage planning) within a novel framework that localizes unknown ROIs with an arbitrary initial position of the robot.The novelties of proposed framework in two folds. Firstly, our approach based on gradient based boundary estimation overcomes the limitation of previous approaches (detail explanation can be found in Section 4). Secondly, inspired by existing area coverage approaches, throughout the results, we demonstrate the performances of our two different algorithms namely Voronoi-based coverage and recursive geometric subdivision.

Although proposed framework is applied in the context of field radiation monitoring, our approach is general and can be scaled to other domains where an opportunistic collection of environmental phenomena is necessary.

7.1 Problem Formulation

We are given a target area T, which contains radiation sources, which strength can be sensed by the robot. We assume that T can be decomposed into a regular grid with n cells. Let us denote this grid by G. Since radiation sources might be spatially distributed, measurement attributes are not available for every cell. Thus, G contains two type of cells, i.e., free cells and contaminated cells. Furthermore, nearby sources cumulatively affect the target area, resulting in a joint distribution of measurement attributes. Let us assume that each cell c is associated with a measurement attribute z. The robot is equipped with a sensor to make a point-wise measurement $z(t)$ at its position $x(t)$ at time t. The Regions of Interest (ROIs) in T are those cells $\mathbf{J} := \cup\exists c$ where the robot finds $z > 0$. The contaminated areas are continuous. Therefore, the robot can trace such areas by tracking only to the boundaries. Therefore, the definitions of the contaminated

and the free cell are quantified through a binary probability value given by

$$p_c = \begin{cases} 0, & \text{if } z \approx 0 \\ 1, & \text{otherwise} \end{cases}. \tag{7.1}$$

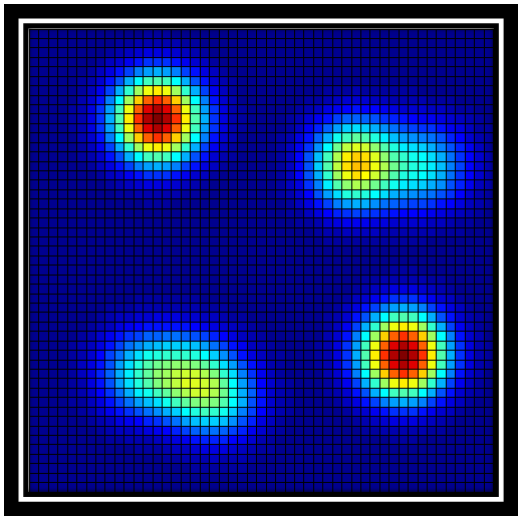


Figure 7.1: The dark blue cells have no measurement attributes whereas other colored cells represent the measurement attributes.

Fig. 7.1 shows an example world map of size 50×50. Depending on the spatial locations of the radiation sources, measurement attributes are also spatially distributed throughout T. The dark blue cells are the cell where $p_c = 0$. The other colored cells represent the fact that measurement attributes are available such that $p_c = 1$. We can then find multiple ROIs while splitting $\mathbf{J}$ subject to spatial distances.

Definition 7.1.1. ***Regions of Interest (ROIs):*** *A collection of cells corresponds to a set of contaminated locations in a given target area T, i.e. the set $\{\mathbf{J} \in T | p_c = 1\}$.*

The global mission of the robot can be defined in two different ways, which implies two different objective functions as follows

- the minimum time to localize an ROI,
- the total time to localize all the ROIs in T.

Without losing any generality, we assume that the travel time is proportional to the travel distance. Therefore, firstly, we will use the boundary estimation technique that minimizes the robot's exploration to localize an ROI. Secondly, we will use the heuristic area coverage technique that ensures to localize all the ROIs in T. The total time is taken into account by summing up the boundary estimation paths and the heuristic area coverage paths.

Let us formally define these objective functions. First, starting from an initial cell, we denote the coverage path followed by the robot throughout the free cells by $\mathcal{P}$. Note that, $|\mathbf{J}| << n$ *i.e.*, the number of contaminated cells are much lesser than free cells. We define the event $S_{\mathcal{P}}$ as the event that the robot reaches to any ROI which is not localized beforehand. The complete coverage path $\mathcal{P}$ can be then discretized by the presence of ROI. Therefore, the expected probability to find an ROI can be expressed as follows

$$E[S_{\mathcal{P}}] = \sum_{c \in \mathcal{P}} (1 - p_c)\,. \tag{7.2}$$

Thus, the first objective is to find an online coverage path that minimizes $E[S_{\mathcal{P}}]$. Note that, in this objective, the heading of the path is not important, once the robot heuristically reaches any location of an ROI, the boundary tracking algorithm is followed to determine the ROI size.

For the second objective, we denote the sequence of newly discovered ROIs along the coverage path $\mathcal{P}$. if there exists k number of ROIs in T, we discretize $\mathcal{P}$ into a subset $Q = \{q_1.q_2, ..., q_k\}$. Since the travel time is proportional to the length of q_k, we want to find the minimum length paths in the set Q to localize all the ROIs. Therefore, the total events $C(\mathcal{P})$ that the robot is experienced to localize a finite set of ROIs given by

$$C(\mathcal{P}) = \sum_{q_k \in Q} S_{q_k} \;\; s.t. \; |Q| \leq |ROI|, \tag{7.3}$$

where $|Q|$ is the cardinality of set Q and $|ROI|$ is the number of ROIs are detected in T. If $|ROI|$ is *a priori* given, our focus is to find the minimum exploration time to achieve that number. We then derive the performance index of the robot from eq. (7.3). A formal definition of the performance index as follows.

Definition 7.1.2. ***Performance Index (PI):*** *The performance index of the robot is evaluated with respect to the minimum explored path to localize all of ROIs, i.e.* $PI = \operatorname{argmin} C(\mathcal{P}) \;\; s.t. \; |Q| \leq |ROI|$.

Since we do not know the exact number of ROIs exists in T, it is not possible to stop the robot's exploration when all ROIs are localized. Then, the robot exploration can be terminated by exploration budget. Otherwise, the robot's task is to plan an online path through T such that every ROIs is rapidly localized while subject to complete area coverage.

7.2 Algorithm Descriptions

Fig. 7.2 shows the overall schematic of our proposed system. The algorithm we propose can be broken down into three steps. In the first step, *Adaptive Hierarchical Area Decomposition*, we adaptively partition the target area in hierarchical order to reduce the search space of the robot. We then find the subregions given by the partition using the *Finding subregions*. When the subregions are determined, we examine the utility to traverse each subregion that explained in the *Utility function design*. The subregion which has maximum utility, we plan a coverage path through the set of unvisited cells. The robot progresses through this path. If the robot notices an ROI along its path, it will drop exploring more and iterates whole steps. Otherwise, the whole steps iterated after traveling along the entire path.

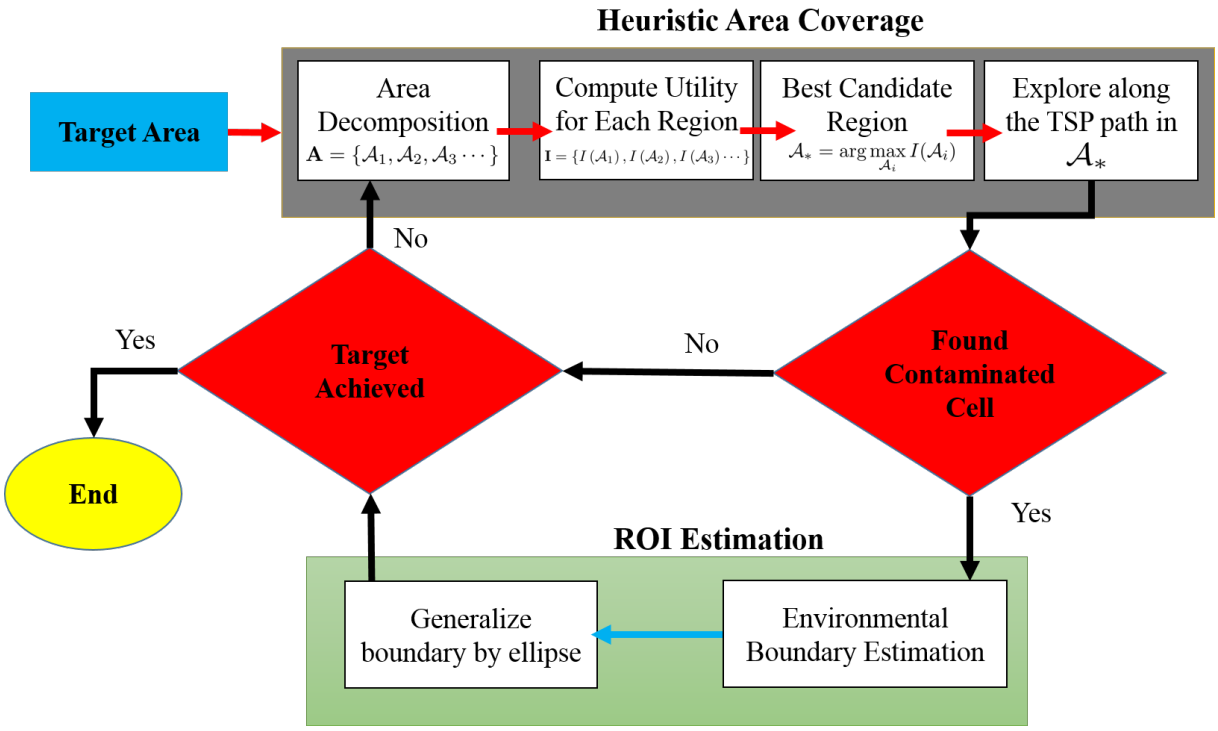


Figure 7.2: **System Overview:** The figure shows all the steps performed by the heuristic area coverage, and ROI estimation algorithms. Starting from an arbitrary location, the robot can iteratively localize the desired number of ROIs using this framework.

7.2.1 Adaptive Hierarchical Area Decomposition

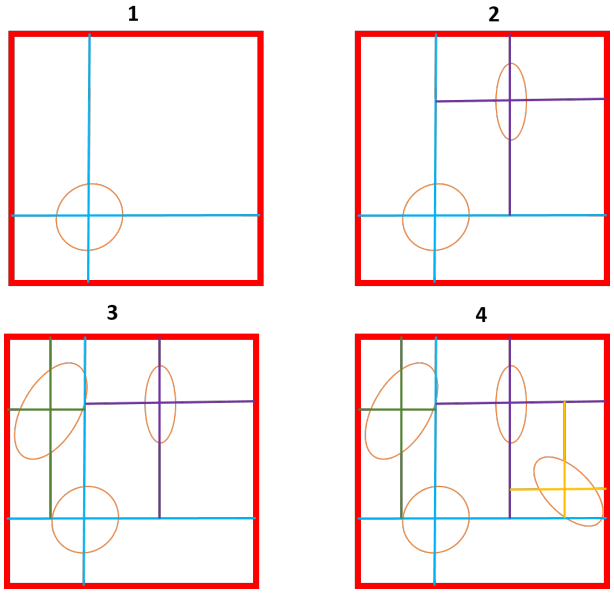
To reduce the computational complexity while navigating a large environment, the search space for the path planning needs to be at a tractable level. We argue that these objectives can be achieved by adaptive partitioning of the target area in hierarchical order. Given the position of ROIs, the hierarchical order is determined by a local

minimum distance with the respect to the robot's relative position. Therefore, we propose the recursive quad division and the Voronoi-based partition in the sense of limiting the search space. Fig. 7.3 shows the overall overview of each algorithm. With a given partition, our goal is to find an ROI through the limited exploration.

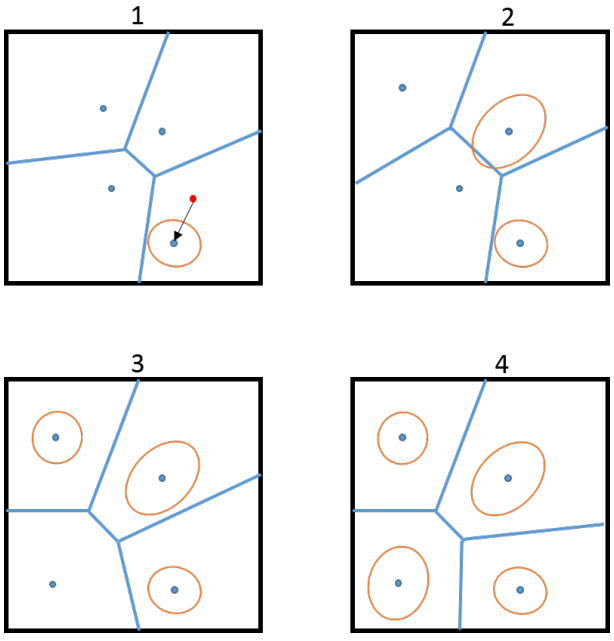
Recursive quadratic subdivision The recursive quadratic subdivision (RQS) algorithm follows a greedy approach, wherein each step it leads the robot to the nearest ROI to its current location that has not been covered yet. The main idea is that initially an optimal path is generated to include every cell in T, which induced from the grid cells. A simple TSP algorithm is adapted to generate such type of path [96], since it minimizes the path length between the robot's current location and all other unvisited cells in the grid. The robot starts to explore along this path. When a contaminated cell found, it switches to the boundary estimation planner. An ROI then computed from the estimated boundary. Thus, the robot determines a minimum route from its location even executing the TSP path entirely. If there are no contaminated cells in the target area, then the robot always optimally explore the whole target area. However, in the presence of multiple ROIs, this coverage path can be further optimized based on a simple heuristic search.

In the second phase, RQS finds a coverage path that minimizes the travel distance to connect the desired number of ROIs. Finding such a path is possible by subdividing the area into four quadrants from the center of ROI. As we iteratively localize the ROIs, the geometric partitions are also recursively subdivided. As a result, if the area turns out to be either the entire target area or a subdivision of the previous decomposition, it is further decomposed into four divisions based on the center of ROI. The three basic operations of this decomposition are as follows. Firstly, we generate a TSP path to explore the unexplored cells optimally. Secondly, when an ROI is determined, we terminate the exploration and decompose the area. Finally, the region of each division is determined by Alg. 9.

We demonstrate the recursive quadratic subdivision while the robot is covering its free space using an example depicted in Fig. 7.3(a). The robot starts to cover the space in a vast cell by generating a TSP path over the target area from the leftmost corner; the target area is shown as the red rectangle in Fig. 7.3 (a). When the robot reaches the cell where $P_c = 1$, which is the unvisited location of a contaminated area, it finishes covering the TSP path. Since the contaminated area is unknown *a priori*, the robot follows the boundary tracking algorithm to cover it. The robot then constructs an ellipse over the estimated boundary to represent the ROI, shown as the orange ellipse in Fig. 7.3 (a). At this point, it encounters the quadratic subdivision at the center point of the ellipse, shown as blue lines in Fig. 7.3 (a). The robot chooses the subdivision



(a) RQS decomposition:
The area is decomposed into four subdivisions based on the center position of ROIs.



(b) VBS decomposition:
Starting with the random partitions, the partitions updated by the center position of ROIs.

Figure 7.3: **Area decomposition:** Two different algorithms are proposed to decompose the search space into smaller regions. The RQS decompose the area in a greedy manner, while the VBS iteratively approaches to optimal decomposition.

that maximizes the utility function, and repeats the step described above as shown in Fig. 7.3 (a). Since the hierarchical quadratic subdivision is connected to each other, the robot is guaranteed to visit all the subdivisions in the target area, and thus completely cover the space.

Voronoi Based Subdivision The Voronoi-based subdivision (VBS) uses the Voronoi-based approach to partition the target area. The main idea is to partition the area by representing the ROI centers as the Voronoi centroids. Since in our case the ROI centers are not apriori available, we have introduced a few changes to the original Voronoi-based partition algorithm. Firstly, it randomly partitions the target area using four random points inside the target area. Secondly, it leads the robot to the nearest centroid from its initial location. Finally, similar to the RQS, TSP algorithm generates the coverage path. The robot starts to explore along this route when a contaminated cell found; it switches to the boundary estimation planner. An ROI then computed from the estimated boundary. Unlike RQS, the robot finds a minimum route to ROI from its location either while traveling to the Voronoi centroid or while executing TSP path. Although these paths increase the probability of finding ROI, if there are no contaminated cells in the subregion, then the complete coverage path would be larger than RQS coverage path because of traveling to the centroid. Note that in VBS, the initial search space limited by the random partition, while in RQS, the initial search space is the whole target area. The partition of the target area updated by the center position of the detected ROI.

In the second phase, VBS finds a coverage path that connects the desired number of ROIs. Finding such path is possible by iteratively updating the Voronoi centroids. The iterative updates of centroids lead VBS to generate an optimal partition of the search space. However, when the number of ROIs is greater than the number of random initial points, the partition centroids are not only iteratively updated but also incrementally constructed. The four basic operations of this decomposition are as follows. Firstly, we generate randomized incremental construction of partitions to reduce the search space. Secondly, the robot moves to the Voronoi centroid, and TSP algorithm creates a coverage path to explore the unexplored cells optimally of a given subdivision. Thirdly, when an ROI is determined, we terminate the exploration and update the Voronoi centroids. Finally, the region of each division is determined by Alg. 9.

We demonstrate the Voronoi-based subdivision while the robot is covering its free space using an example depicted in Fig. 7.3(b). Voronoi Diagram is the partitioning method of a plane with n points into a specific subset of the plane such that each subset contains exactly one generating point. In typical Voronoi diagram, the set of generating points is *apriori* known. The Voronoi polygons are then constructed such that every

Algorithm 9 Finding subregions

Require: Graph, $G = (V, E, B)$
Ensure: Subregions, Λ
for all $e \in E$ **do**
  $\psi_G \leftarrow intersect(e, B)$
  $E_\psi \leftarrow trim(e, \psi_G)$
end for{Shorten initial edges}
for all $b \in B$ **do**
  $\psi_b \leftarrow intersect(b, E)$
  $E_b \leftarrow combination(\psi_b, 2)$
  $E_b \leftarrow unique(E_b)$
end for{Finding box edges}
$E \leftarrow \{\{E_b\} \cup \{E_\psi\}\}$ {update graph}
for all $p \in \psi_c$ **do**
  $\Lambda \leftarrow \cup NeighborEdges(p, E)$
end for{update partition area}

point in a given polygon is closer to its generating point than to any other. However, in our case, we randomly initialize the generating points and iteratively update their positions.

The robot starts to cover the space in a vast cell by moving into the centroid of the current Voronoi region (red dot) which is located at the rightmost corner; the target area is shown as the black rectangle in Fig. 7.3 (b). Then, the robot constructs a TSP path to cover the given region. Whenever the robot reaches the cell where $P_c = 1$, which is the unvisited location of a contaminated area, it finishes covering the centroid path or the TSP path. Since the contaminated area is unknown *a priori*, similar to the RQS, the robot follows the boundary tracking algorithm to cover it. The robot then constructs an ellipse over the estimated boundary to represent the ROI, shown as the orange ellipse in Fig. 7.3 (b). At this point, it encounters the update of Voronoi centroid. The Voronoi centroid of the current region is replaced by the center point of the ellipse, shown as blue dots in Fig. 7.3 (b). If there are more ROIs than the Voronoi centroids which are chosen initially, the overall Voronoi partitions are reconstructed with updated centroids. Note that the minimum number of subdivisions in this case is four, and the algorithm can also cover more than four subdivisions. The robot chooses the subdivision that maximizes the utility function and repeats the step described above as shown in Fig. 7.3 (b). Since the Voronoi regions are connected, the robot is guaranteed to visit all the subdivisions in the target area, and thus completely cover the space.

7.2.2 Finding subregions

At the end of the second phase, each algorithm finds the subregions based on its partition method. For this purpose, it begins by creating the graph $G = (V, E, B)$ induced from above mentioned methods. We represent the target area as a rectangular box B in G. The initial partitions are the edge set E that includes edges with infinite lengths. To find subregions Λ, firstly, we shorten each edge $e \in E$ subject to B. Let V be the set of vertices that includes three types of subsets such that $V = \{\{\psi_G\}, \{\psi_b\}, \{\psi_c\}\}$. Let ψ_G be the first subset of V that represents the vertices at the intersection between B and E. Also, let ψ_b be the set of vertices that represents the corner points of B, and let ψ_c be the centroid of ROIs. Once we trim the long edges, the new partition represented by E_ψ. Secondly, we find all the possible combination of edges on B and represent by E_b. The G is then updated by combining these two set of edges such that $E \leftarrow \{\{E_b\} \cup \{E_\psi\}\}$. Finally, we group all subregions Λ by finding the neighbor edges. Finding such a neighbors is straightforward. Given ψ_c, an anti-clockwise walk along the E can sort such neighbors.

7.2.3 Utility function design

In the third phase, each algorithm finds the best search space among all subdivisions of the target area. For this action, it computes the utility between each of subdivisions. The utility is designed to favor destinations which offer higher expected information gain. Throughout this work, we use an explored grid map, m, to model the environment. This map is a binary map where each cell represents visited or unvisited information. Let i be the index of each subdivision and the division of such a map satisfies the following equation

$$m = \sum_i m^{[i]}. \tag{7.4}$$

An action a_t generated at time step t is represented by a sequence of relative movements $a_t = \hat{u}_{t:T-1}$ which the robot has to carry out starting from its current position x_t. During the execution of a_t, if the robot finds a contaminated cell along its path, then it estimates an ROI in the map. Therefore, the explored trajectory of the robot indicates some of the cells in m as follows

$$x_{1:t} = \exists c \in m. \tag{7.5}$$

In the case when the robot finds an ROI in the map, we have to treat the ROI cells differently. We assumed that traveling inside an ROI is redundant, and want to avoid such a region. Therefore, the cells bounded by an ROI considered as similar as visited

cells. Let d_t be the set that represents these cells as follows

$$d_t = \{\forall c \in ROI1, \forall c \in ROI2 \cdots\}. \tag{7.6}$$

Assuming that each cell c in m is independent of each other. Then the posterior entropy of m can be computed as follows

$$\begin{aligned} H(p(m|x_{1:t}, d_t)) = -\sum_{c \in m} p(c) \log p(c) + \\ (1 + p(c)) \log(1 - p(c)). \end{aligned} \tag{7.7}$$

Given a subdivision, since the robot does not know when it will find an ROI along its path, the coverage path should include all cells to compute the expected information gain. Thus, the entropy of target subdivision can write as follows

$$\begin{aligned} H(p(m^{[}i]|x^{[i]}_{t+1:T}, d_t, a_t)) = -\sum_{c \in m^{[i]}} p(c) \log p(c) + \\ (1 + p(c)) \log(1 - p(c)). \end{aligned} \tag{7.8}$$

To compute the information gain of a subdivision, we calculated the change in entropy caused by the integration of posterior and predicted prior into the robot's world model as follows

$$I(m^{[i]}, a_t) = H(p(m|x_{1:t}, d_t)) - H(p(m^{[i]}|x^{[i]}_{t+1:T}, d_t, a_t)). \tag{7.9}$$

After computing the expected information the utility for each action under consideration, we select the action a^*_t with highest expected information

$$a^*_t = \arg\max_{a_t} I(m^{[i]}, a_t). \tag{7.10}$$

There are some works in exploration and mapping problems that consider another quantity besides the information gain in Eqn. (7.10). That is the cost to reach the subdivision. However, we observed that adding such a quantity with the utility function decreases the overall performance of both algorithms. Thus, every time the robot has to make the decision where to go next, it uses only information maximization metric to determine the action a^*_t.

7.3 Finding ROIs

We employ a boundary estimation algorithm to determine the ROIs by using the proposed exploration method. In this section, first, we will explain how to generalize an

arbitrary boundary. Next, we will show the generalization properties. Since the ROIs are the function of boundaries, we then focus on only the analysis of boundary estimation.

7.3.1 Environmental Boundary Generalization

ROIs over the target area T are dependent on the boundary line estimated by environmental boundary algorithm. Memorizing a complex boundary is computationally expensive, therefore to obtain the tractable level of computation, we require the parametric estimation of the boundary.

Definition 7.3.1. ***Boundary line:*** *The line is said to be boundary line if it represents the intersection between the contaminated area and non-contaminated areas.*

Assume an contaminated area $\delta\mathcal{A}$ is a non-convex set where the continuous boundary is defined by a level set $\delta\mathcal{A}$ as follows

$$\delta\mathcal{A} = \left\{ x \in \mathbb{R}^2 | z(x) = \beta \right\}, \tag{7.11}$$

where β is the measurement threshold.

Boundary algorithm ensures that an environmental boundary can be estimated by tracking the robot states such that $\delta\mathcal{A} = \{x_{1:t}\}$. When the exploration is terminated, this set $\delta\mathcal{A}$ can be used to estimate of the best fit to an ellipse. This generalization is done by the least squares criterion from the set $\delta\mathcal{A}$. We also consider the possible tilt of the ellipse from the conic ellipse representation as follows

$$ROI(\delta\mathcal{A}) = a{S_x}^2 + bS_xS_y + c{S_y}^2 + dS_x + eS_y + f = 0, \tag{7.12}$$

where $\{S_x, S_y\} \in \delta\mathcal{A}$ and a, b, c, d, e, f are the parameter for a second degree polynomial equation. After the estimation, the tilt is replaced by a rotation matrix from the ROI, and then the rest of parameters are extracted from the conic representation.

7.3.2 Analysis of Boundary Estimation Algorithm

Proposition 7.3.1. *Given a contaminated area $\mathcal{A}_i \in \mathbf{A}$, the measurement attribute of a location x is not available if and only if $x \notin \mathcal{A}_i$.*

Proof. Assume that the cardinality of $\mathbf{A}$ is 1 for a given T. To prove this proposition, first we will show that all the locations bounded by the boundary $\delta\mathcal{A}_i$ are important and all the outside locations from boundary are negligible.

The sufficiency of this proposition is trivial and we can easily prove it by Definition 1. By Definition 1, $z(x) > \beta$ if x is the close proximity to the sources. By Definition 3,

we can say that all the sources are covered by the boundary lines. Since β represent the boundary threshold, we can conclude that $\forall x \in \mathcal{A}_i$ where $z > \beta$ are bounded by $\delta\mathcal{A}_i$. Similarly, the necessity of this proposition also follows the Definition 1. Let, x_h is the hotspot location for $\mathcal{A}_i$ where $z(x_h)$ maximum subject to $\forall x \neq x_h$. As the robot travel far from x_h the following relationship hold by Definition 1, $z(x_h) > z\,(\forall x \neq x_h)$. When $z(x) << z(x_h)$ *i.e.,* the measurement of x is too less than the x_h, we neglect those locations by drawing boundary line. Therefore, for any location outside the boundary line, the measurement is $z(x) < \beta$ and it is negligible. □

Proposition 7.3.2. *Let $\delta\mathcal{A}_i$ be the estimated area generated by the boundary tracking boundary algorithm. Then $\mathcal{A}_i$ is a contaminated area if the effects of every nearby sources can be jointly described by the area shape.*

Proof. This is true by the construction of Eqn. (7.11). In particular, since $\delta\mathcal{A}_i$ is continuous, then the contaminated area $\mathcal{A}_i$ must be continuous. Thus, the continuous $\mathcal{A}_i$ represents the joint effects of every near by sources. □

It is evident from the above proposition 7.3.2 that the boundary tracking algorithm can estimate an ROI. Once boundary tracking algorithm computes the environmental boundary set $\delta\mathcal{A}$, we determine the ROI by fitting an ellipse to the sample points of $\delta\mathcal{A}$. This generalization has done by minimizing the least square distance between the sample points and the Conic representation of the ellipse. After that, we extract the parameters from that Conic representation by removing the tilt. Next, we will explain how to extend this approach to find multiple ROIs.

Lemma 7.3.1. *Given a target area T, if $\delta\mathcal{A}_1$ and $\delta\mathcal{A}_2$ are the two boundaries. Then $\delta\mathcal{A}_1 \cap \delta\mathcal{A}_2 = \emptyset$.*

Proof. Suppose $\delta\mathcal{A}_1 \cap \delta\mathcal{A}_2 \neq \emptyset$. Then the contaminated area $\mathcal{A}_1$ be overlapped with the $\mathcal{A}_2$, which contradicts the proposition 7.3.2. □

The above lemma 7.3.1 implies that the boundary tracking algorithm can efficiently separate the contaminated areas.

Theorem 7.3.1. *Given a target area T, the boundary algorithm can estimate the boundary of every contaminated area, i.e. $\forall\delta\mathcal{A}_i \in \mathbf{A}$.*

Proof. Suppose there are n number of contaminated areas in T such that $n = |\mathbf{A}|$. We use mathematical induction to prove this theorem.

- Base case: if $n = 1$, then it is trivial that the boundary algorithm can uniquely estimate the boundary of contaminated area $\delta\mathcal{A}_i$ without overlapping any others.

- Induction step: Suppose $n = 2$. Then there are two cases when estimated boundaries are not equal to n. Firstly, estimated boundaries are greater than n is impossible, because it contradicts definition 7.3.1. Secondly, it is also not possible that estimated boundaries are less than n, because it contradicts lemma 7.3.1. Therefore, estimated boundaries are exactly equal to n, as required.

□

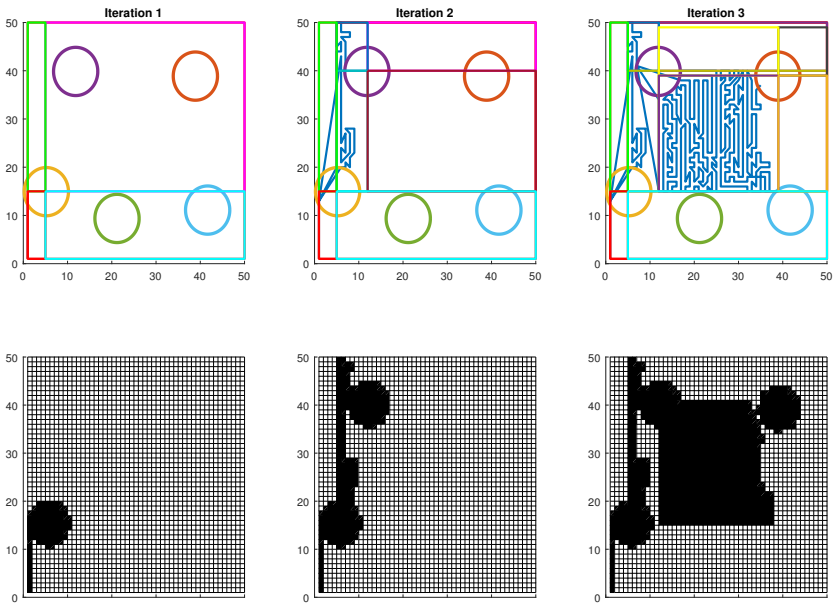
7.4 Simulation Results

To find the shortest coverage path, we perform 4 different experiments. We assume that the target area contains at most 5 ROIs. The performance of each algorithm was evaluated by the distance of coverage paths. To demonstrate the efficiency, we start localizing 2 out of 5 ROIs and conclude by 5 out of 5 ROIs. We also have analyzed the worst case performance and we present a statistical analysis of two algorithms from 20 trial runs. The performance of algorithms significantly varied from each other. In particular, we have observed a noticeable difference of the algorithms on localizing uniformly distributed random ROIs. It is noteworthy that to compute the efficiency, the ROIs shape should remain fixed for each algorithm, we then overlook the additional path cost required to estimate ROIs.

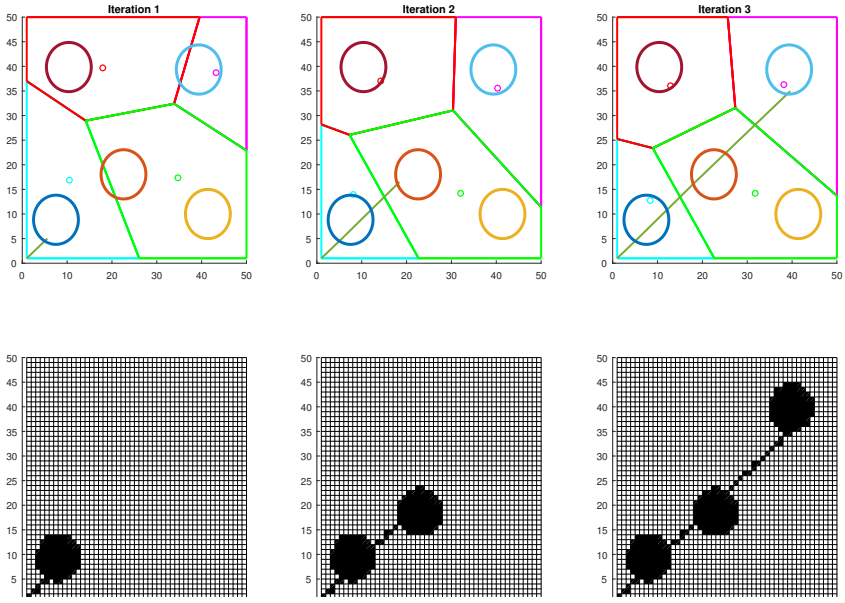
7.4.1 Finding coverage path that connects the desired number of ROIs

We now consider the case of finding ROIs that meet the desired level of exploration. Therefore, we focus on the shortest coverage path for a given number of ROIs. We consider a $50m \times 50m$ grid area where 5 uniformly distributed random ROIs are located. Starting from an initial location $(1, 1)$, the robot has to find the minimum coverage path that connects the desired number of ROIs. The coverage path can be found by adjusting the cost to the inversely proportional to the unexplored area. In another word, the robot explores the mostly unexplored region first.

Fig. 7.4 (a) shows a toy example of RQS algorithm. In RQS, the initial search space is fixed to the whole target area. Once an ROI is found in a subregion, the search space is subdivided into four regions based on the center position of the detected ROI. The robot avoids exploring the cells bounded by the ROI and starts its exploration from the nearest corner position of a new subregion. Next time only that subregion is divided into four more divisions again. These processes are iterated until the end of the mission. We have observed that the smaller the search space and the more efficient the RQS is.



(a) RQS coverage paths on a sample map with uniformly distributed random ROIs. The blue line in upper figure shows the coverage path, while the colored lines are the partition of the target area. A region is divided into 4 subregions based on the center of ROI. For a new region, the searching process is started from the corner. The search spaces are iteratively reduced based on the position of the center of ROIs.



(b) VBS coverage paths on a sample map with uniformly distributed random ROIs. The dark green line in upper figure shows the coverage path, while the colored lines are the partition of the target area. The centroid of each region is represented by the same colored cycle. For a new region, the searching process is started from the centroid. The partitions are iteratively updated based on the true position of the center of ROIs.

Figure 7.4: **Coverage path:** The robot starts the coverage in cell (1,1) and detects any 3 ROIs out of 5. The shape of each ROI is elliptical and is represented in unique color. The lower grid map represents the coverage map. The measured cells are represented by black color. A cell is called to be measured if it is included either in coverage trajectory or it is bounded by the detected ROIs. In general, the VBS's coverage path is shorter than the RQS's.

A basic reason is that the complexity of TSP algorithm increases with the dimension of search space. However, the RQS has a property to sequentially narrow down the search space, resulting in faster convergence when the ROIs are closely located. On the other hand, this kind of heuristic subdivision may cause the increment of the traveling time of the robot from its current position to the unexplored region.

Fig. 7.4 (b) shows a toy example of VBS algorithm. In VBS, the initial search space is generated by randomly choosing 4 points bounded in the target area. We will call these points as the Voronoi centroids. The initial search space is then subdivided into four regions based on the Voronoi centroids. The robot moves the centroid of a Voronoi region first and exhaustively search for an ROI within that region. When an ROI is found whether traveling to the centroid or searching the entire subregion, the robot updates the Voronoi diagram. Similar to the RQS, the robot avoids exploring the cells bounded by the ROI. These processes are iterated until the end of the mission. We have observed that due to the initial smaller search space and the VBS is more efficient than RQS. However, the VBS requires at least 3 points to partition the entire search space optimally. When there are less than 3 ROIs in total area and the robot has to localize all of them, the VBS performance is not stable as compared to the RQS.

7.4.2 Performance comparison

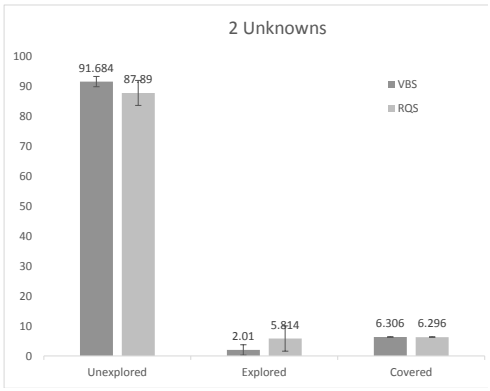
Fig. 7.5 shows a performance comparison. To access the long-term performance of each algorithm, we ran the same experiments for 20 times by gradually increase the target numbers. Fig. 7.5 (a), (b), (c), (d) shows the results in area coverage percentage metric. We divided the given target area into three different regions- 1) explored by the robot 2) covered by the ROIs 3) remained unexplored. Our goal is to minimize the explored region as small as possible. To make a fair comparison, we use five uniformly distributed random ellipses and try to find the shortest path that connects 2, 3, 4, and 5 ROIs. For them, the covered regions by ROIs are 6, 9, 13, and 16 percentage of the target area. The unexplored region then determined by subtracting the covered and explored regions from the total area.

The reduction of search space directly influences of the explored areas. When the number target of ROIs is less than total ROIs existed in the target area, the robot dramatically reduces the amount of explored region. In worst case scenario, when the robot needs to localize all five ROIs, it requires traveling more locations to find the ROIs, resulting in higher exploration regions. However, the performance of each algorithm is not stable, and we use the error bar of the bar chart to represent their standard deviation. For both algorithms, the deviation increases with the increment of the number of target ROIs.

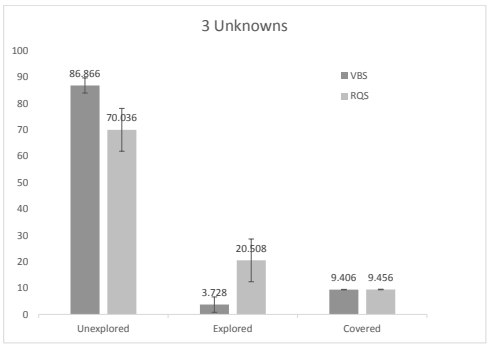
It is evident from the Fig. 7.5, the VBS always outperforms the RQS because of the optimal search space division strategy. Furthermore, when the number of target ROIs is less than the total number of ROIs, the VBS significantly reduces the explored region than the RQS. We reported the numeric performance comparison between the VBS and the RQS in Fig. 7.5.

7.5 Summary

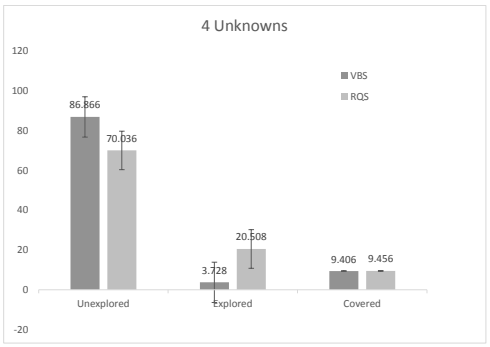
In this chapter, we present a framework to solve the problem of rapidly determining regions of interest (ROIs) in unknown intensity distribution, especially in radiation fields. The vast majority of existing literature on robotics area coverage does not report the identification of ROIs. In a radiation field, ROIs limit the range of exploration to mitigate the monitoring problem. However, considering the limited resources of Unmanned Aerial Vehicle (UAV) as a mobile measurement system, it is challenging to determine ROIs in unknown radiation fields. Given the target area, we attempt to plan a path that facilitates the localization of ROIs with a single UAV, while minimizing the exploration cost. To reduce the complexity of exploration of large scale environment, initially whole areas are adaptively decomposed by two hierarchical methods based on recursive quadratic subdivision and Voronoi based subdivision. Once an informative decomposed sub area is selected by maximizing a utility function, the robot heuristically reaches to contaminated areas and then a boundary estimation algorithm is adopted to estimate the environmental boundaries. This boundary estimation algorithm should have specific properties that are deemed to be incorporated into the proposed framework, therefore, those properties are theoretically analyzed in this chapter. Finally, the detailed boundaries are approximated by ellipses, called the ROIs of the target area and whole procedures are iterated to sequentially cover the all areas. The simulation results demonstrate that our framework allows a single UAV to efficiently and explore a given target area to maximize the localization rate of ROIs.



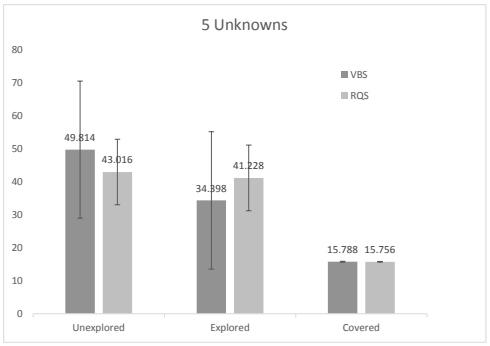
(a) In this experiment, the robot localizes 2 out of 5 ROIs.



(b) In this experiment, the robot localizes 3 out of 5 ROIs.



(c) In this experiment, the robot localizes 4 out of 5 ROIs.



(d) In this experiment, the robot localizes 5 out of 5 ROIs.

Figure 7.5: **Area coverage:** The performance is evaluated by comparing the size of following areas- unexplored, covered and explored area. The error bar represents the standard deviation of each area.

Chapter 8

Conclusions and Future work

8.1 Thesis Contributions

The approaches presented in this thesis represent different types of path planning strategies that tackle the source localization problem in the radiation field. Focusing on the limited flight time of UAV and computational resources, we have subdivided the main issue into four subproblems. Our approaches strive for the optimal solution when mission time is limited and heuristic approximation when computational resources are restricted. Finally, we show how to implement such path planning algorithms in a real robot platform.

The main contributions of this thesis are as follows:

Hotspot Localization In chapter 3, we have proposed an online HexTree path planner for UAV navigation to efficiently localize the radiation hotspot. The sampled path generated by the HexTree planner was optimal and ensured termination of exploration. After localizing the hotspot, a loop closing trajectory was generated from sampled trajectories, which contained the most informative locations considering the travel cost. Through our theoretical analysis, we have proven the optimality of the HexTree path and found an upper bound of path length. Since the UAV explored a large area for spatial sampling, the HexTree planner clearly outperformed the RIG-tree planner in terms of the distance traveled and the convergence time. The contributions of this chapter are as follows:

1. A novel hexagon tiling based workspace decomposition,
2. Efficient HexTree based sampling strategy,
3. Optimal return path generation strategy, and
4. Theoretical and numerical analysis.

Boundary Estimation In chapter 4, we presented an improved PID control law that derives a UAV to track the boundary of unknown environments. On one hand, the integration of EKF in conventional framework ensures an accurate prediction even in the presence of the noisy sensor reading. On the other hand, adaptive crossing angle correction eliminates the extra burden on controller tuning. We showed that the robot can accurately estimate the boundary by adding more samples in non-concave regions. Finally, we proved how to estimate a boundary in a finite time limit.

Three different simulation scenarios were considered to evaluate the efficiency of the proposed algorithm. The tuning parameters remain unchanged in all experiments, the results of similarity analysis show that the proposed modifications outperform others in all the cases.

The contributions of this chapter are as follows:

1. We have formulate the boundary estimation problem which does not require *a priori* information at all.

2. Our algorithm can estimate the boundary in a fast manner while minimizing the exploration of UAV.

3. The proposed algorithm is complete, which means that the estimation process terminates within a finite operation time.

4. Focusing on the limited computational capabilities of the UAV, the proposed algorithm can robustly determine the boundary.

Source Localization A single UAV exploration based multiple unknown radiation sources localization problem is investigated in chapter 5. Three different cases of spatial distributed sources are considered to demonstrate the efficiency of the proposed algorithms. In order to explore a large radiation field using a UAV, we propose to adopt a topographic mapping strategy along with the reduction of ROI in the field. The segmentation of a large radiation field was primarily done by a novel log-gradient classifier (lgc) that segregates a priori known trajectory coupled with measurements. The trajectory connects the lower intensity region to the hotspot region of the field, where the lgc segments it into a finite number of key positions. The contour lines are then generated by tracking a constant intensity value. However, the similarity analysis of contour shape indicates that the ROI in an unknown radiation field can be further reduced by avoiding superfluous contour lines.

In order to localize the radiation sources, the mitigation strategy such as proposed framework is demonstrated to be effective. It can easily deal with the trade-off between the cost of robotic exploration and the accuracy of source localization. Reducing the

ROI could potentially be effective in making the robot aware of sources positions. The diagnostic criterion used for analyzing the ROI shape can be extended to predict the type of distributed sources. Although the radiation sources might be arbitrarily located in an unknown radiation field, the proposed framework not only accelerates the mission completion time but also leads to accurate estimation close to actual sources. In the numerical simulation, it can be seen that with the proposed ROI selection method, the VB clearly outperforms the HT in terms of accuracy. To determine the clustered sources, the HT can easily show a as similar performance as the VB without any additional exploration. Thus, the proposed framework opted the HT only for the clustered cases, otherwise, the VB is adopted.

The contributions of this chapter are as follow:

1. Characterizing the cumulative radiation effects with multiple sources.
2. Finding the region of interest **(ROI)** in large radioactively contaminated areas to narrow down the search area.
3. Quickly and accurately localizing the sources that are actively acting in the radiation field.

ROIs Determination In this chapter, we have discussed the ROIs determining problem for a large environment and its various aspects. First, we have proposed a novel online framework to integrate the environmental boundary estimation and area coverage problems. Second, we theoretically analyze the properties of the boundary estimation algorithm which is deemed to best satisfy such conflicting requirements. Third, we proposed two different adaptive area decomposition and search algorithms to localize the desired number of ROIs rapidly: RQS, which uses a greedy-based approach for reducing the search space, and VBS, uses an optimal partitioning strategy for updating the search space. Fourth, we demonstrate these algorithms in a simulated environment, and statistically analyze their relative performance.

The simulation results show that, in general, VBS creates coverage path is shorter than the coverage path by RQS. VBS has clear benefit when handling fewer ROIs since it performs a global planning of the coverage according to the size of the target area. On the other hand, RQS plans only local best decomposition, resulting in overall poor performance. Both algorithms do not require to complete coverage of the target area and save a significant amount of redundant exploration. Comparing all the experiments, we have shown that, in general, required explored areas are less than unexplored areas. Furthermore, the robot does not need to visit the covered areas by ROIs. As a result, even in worst case scenarios, the required exploration to determine ROIs is always less than complete area coverage algorithms.

In future, we would like to extend the algorithms for multi-robot systems. We would also consider the problem associated with non-stationary environmental boundaries.

The contributions of this chapter are as follows:

1. We have formulated the localization of ROIs which does not require *a priori* information at all.

2. Our algorithm can localize ROIs in a fast manner by minimizing the exploration of UAV.

3. The proposed algorithm is complete, which means all contaminated locations are identified for the long operation of UAV.

4. Focusing on the limited computational capabilities of the UAV, proposed algorithm can robustly determine ROIs.

5. To our best knowledge, this is the first approach that integrates the environmental boundary estimation problem to the area coverage problem.

Implementation We elaborate the implementation details in Chapter 7. The real world demonstrations of such path planning algorithms are motivated by implementing a trajectory controller with an accurate state estimator in a low-cost aerial platform. We remove the assumption of GPS-based localization and discuss how a UAV could navigate in an unknown and GPS-denied environment.

8.2 Future Directions

3D Path Planning One dimension we did not include this work is the consideration of three dimensions (3D) path planning, which is essential for localizing radioactive sources. While maintaining constant altitude may still result in plausible plans, adding z-axis planning may increase the localization accuracy. For example, measuring a location by changing altitude may provide a better estimation since the real world is 3D.

One way to incorporate this constraint is to integrate 3D search algorithm to the path planner. This embedding will allow us to gather 3D sensor reading from the environment. The cost of the path should be further revised by including the travel cost along the z-axis.

Since the increment of one dimension in search space significantly increase the computational complexity, the 3D search may not perform well in real time. Therefore, investigating this and other ways to optimize the search space during the planning process is a subject of future work.

Multi-robot Coverage The ideas presented so far have focused on developing approaches to tackle path planning problems of a single UAV. While a single UAV can cover only certain target areas, others demand to divide the large target area and include multiple robots. For example, suppose we have a group of robots that is assigned the task of searching for radiation sources in a large environment. This problem can be modeled as multi robots coverage problem, highlighting the fact that many tasks related to "ROIs determination" can unite multiple robots routing problems.

The focus for future work is on developing an integrated approach that addresses problems of determining ROIs with multiple robots. The goal is to minimize the mission time while ensuring complete, accurate and computationally efficient determinations.

Non-myopic Observations For the robot exploration, we assumed that no map is available in advance. For our experiments, we only simulated static radiation fields and the robot needs to sense the environment in a point-wise fashion. Since the environment is completely unknown to the robot, it cannot make a good prediction. As the robot sequentially gathers observations, the prediction converges to the accurate result.

For future work, we will make an assumption that a preliminary map is available. Thus, the goal of exploration would be to reduce map uncertainties. Additionally, we will consider the UAV's constraints to design an efficient and effective path planner. Finally, a more thorough evaluation of this algorithm requires testing it in real-world scenarios with different datasets. These experiments will support new issues that need to be addressed for this approach.

Real World Constraints For this work, we made an assumption that the radiation field behaves like a static environment due to limited mission time. However, in certain cases, the environment may be dynamic. The wind disturbance or radiation decay may deviate the radiation field model from its ideal form. In these cases, as the robot explores the environment, the measurements are updated not only in the spatial domain but also in the temporal domain. Furthermore, during path planning in geometric space, we don't consider any obstacles that may hinder to sample in certain locations. Currently, we are not addressing those problems, but this is an interesting research direction for future work.

Bibliography

[1] Shun-ichi Azuma, Mahmut Selman Sakar, and George J Pappas. Stochastic source seeking by mobile robots. *IEEE Transactions Automatic Control, 2308-2321*, 2012.

[2] Nima Ghods, Paul Frihauf, and Miroslav Krstic. Multi-agent deployment in the plane using stochastic extremum seeking. In *IEEE Conf. Decision and Control, 5505-5510*, 2010.

[3] Nikolay A Atanasov, Jerome Le Ny, and George J Pappas. Distributed algorithms for stochastic source seeking with mobile robot networks. *ASME Journal Dynamic Systems, Measurement, and Control*, 2015.

[4] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *Int'l Journal Robotics Research*, 2014.

[5] Shuai Li, Yi Guo, and Brian Bingham. Multi-robot cooperative control for monitoring and tracking dynamic plumes. In *IEEE International Conference on Robotics and Automation*, pages 67–73, 2014.

[6] Muhammad Fahad, Nathaniel Saul, Yi Guo, and Brian Bingham. Robotic simulation of dynamic plume tracking by unmanned surface vessels. In *IEEE International Conference on Robotics and Automation*, pages 2654–2659, 2015.

[7] Abdullah Al Redwan Newaz, Sungmoon Jeong, Hosun Lee, Hyejeong Ryu, Nak Young Chong, and Matthew T. Mason. Fast radiation mapping and multiple source localization using topographic contour map and incremental density estimation. In *IEEE International Conference on Robotics and Automation*, pages 1515–1521, 2016.

[8] Jerry Towler, Bryan Krawiec, and Kevin Kochersberger. Terrain and Radiation Mapping in Post-Disaster Environments Using an Autonomous Helicopter. *Remote Sensing*, 4:1995–2015, 2012.

[9] BA White, Antonios Tsourdos, I Ashokoraj, S Subchan, and Rafal Zbikowski. Contaminant cloud boundary monitoring using uav sensor swarms. *AIAA Journal of Guidance, Control, and Dynamics*, 2005.

[10] David W. Casbeer, Derek B. Kingston, Randal W. Beard, and Timothy W. McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *Int. J. Systems Science*, 37:351–360, 2006.

[11] Balakrishna Gokaraju, Surya S. Durbha, Roger L. King, and Nicolas H. Younan. Sensor web and data mining approaches for harmful algal bloom detection and monitoring in the gulf of mexico region. In *IEEE International Geoscience & Remote Sensing Symposium*, pages 789–792, 2009.

[12] Alexey S. Matveev, Hamid Teimoori, and Andrey V. Savkin. Method for tracking of environmental level sets by a unicycle-like vehicle. *Automatica*, 48:2252–2261, 2012.

[13] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 1344–1350. IJCAI/AAAI, 2013.

[14] Tairen Sun, Hailong Pei, Yongping Pan, and Caihong Zhang. Robust adaptive neural network control for environmental boundary tracking by mobile robots. *International Journal of Robust and Nonlinear Control*, 23:123–136, 2013.

[15] Mong-ying A Hsieh. Stabilization of Multiple Robots on Stable Orbits via Local Sensing Stabilization of Multiple Robots on Stable Orbits via Local Sensing. (April):2312–2317, 2007.

[16] Environmental boundary tracking and estimation using multiple autonomous vehicles. *Proceedings of the IEEE Conference on Decision and Control*, pages 4918–4923, 2007.

[17] Jerry Towler, Bryan Krawiec, and Kevin Kochersberger. Radiation mapping in post-disaster environments using an autonomous helicopter. *Remote Sensing, 1995-2015*, 2012.

[18] Mark R Morelande and Alex Skvortsov. Radiation field estimation using a gaussian mixture. In *Information Fusion, Int'l Conf. on, 2247-2254*, 2009.

[19] Ling Xu. Graph Planning for Environmental Coverage. *Carnegie Mellon University*, (August):135, 2011.

[20] Liam Paull, Carl Thibault, Amr Nagaty, Mae Seto, and Howard Li. Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle. *IEEE Transactions on Cybernetics*, 44:1605–1618, 2014.

[21] R. Yehoshua, N. Agmon, and G. A. Kaminka. Robotic adversarial coverage of known environments. *The International Journal of Robotics Research*, pages 1–26, 2016.

[22] Daniel E Soltero, Mac Schwager, and Daniela Rus. Decentralized path planning for coverage tasks using gradient descent adaptive control. *The International Journal of Robotics Research*, 33:401–425, 2014.

[23] Alexey S. Matveev, Michael Colin Hoy, Kirill Ovchinnikov, Alexander Anisimov, and Andrey V. Savkin. Robot navigation for monitoring unsteady environmental boundaries without field gradient estimation. *Automatica*, 62:227–235, 2015.

[24] P. Dames and V. Kumar. Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. *IEEE Transactions on Automation Science and Engineering*, 12:850–864, 2015.

[25] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Int'l Conf. Autonomous Agents and Multiagent Systems, 23-30*, 2008.

[26] Kian Hsiang Low, John M Dolan, and Pradeep K Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Int'l Conf. Automated Planning and Scheduling*, 2009.

[27] Jerry Towler, Bryan Krawiec, and Kevin Kochersberger. Radiation mapping in post-disaster environments using an autonomous helicopter. *Remote Sensing, 1995-2015*, 2012.

[28] Mark R Morelande and Alex Skvortsov. Radiation field estimation using a gaussian mixture. In *Int'l Conf. Information Fusion, 2247-2254*, 2009.

[29] Jren-Chit Chin, David KY Yau, and Nageswara SV Rao. Efficient and robust localization of multiple radiation sources in complex environments. In *IEEE Conf. Distributed Computing Systems, 780-789*, 2011.

[30] Er-Wei Bai, Kidane Yosief, Soura Dasgupta, and Raghuraman Mudumbai. The maximum likelihood estimate for radiation source localization: Initializing an iterative search. In *IEEE Conf. Decision and Control, 277-282*, 2014.

[31] Benjamin Charrow, Vijay Kumar, and Nathan Michael. Approximate representations for multi-robot control policies that maximize mutual information. In *Robotics: Science and Systems*, 2013.

[32] Gregory Hitz, Alkis Gotovos, Francois Pomerleau, Marie-Eve Garneau, Cedric Pradalier, Anna Krause, and Roland Y Siegwart. Fully autonomous focused exploration for robotic environmental monitoring. In *IEEE Int'l Conf. Robotics and Automation, 2658-2664*, 2014.

[33] Philip Dames, Mac Schwager, Vipin Kumar, and Daniela Rus. A decentralized control policy for adaptive information gathering in hazardous environments. In *IEEE Conf. Decision and Control, 2807-2813*, 2012.

[34] RA Cortez and HG Tanner. Radiation mapping using multiple robots. In *ANS Joint Topical Meeting on Emergency Preparedness and Response and Robotic and Remote Systems, 157-159*, 2008.

[35] Redwan Newaz, Abdullah Al, Sungmoon Jeong, Hosun Lee, Hyejeong Ryu, Nak Young Chong, and Matthew T. Mason. Fast radiation mapping and multiple source localization using topographic contour map and incremental density estimation. In *IEEE Int'l Conf. Robotics and Automation, 1515-1521*, 2016.

[36] Jinlu Han and YangQuan Chen. Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field. *Jour. Intelligent & Robotic Systems, 323-332*, 2014.

[37] Inyoung Ko, Beobkyoon Kim, and Frank Chongwoo Park. Randomized path planning on vector fields. *Int'l Journal Robotics Research, 1664-1682*, 2014.

[38] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI Conf. Artificial Intelligence, 1650-1654*, 2007.

[39] Nannan Cao, Kian Hsiang Low, and John M Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Int'l Conf. Autonomous Agents and Multi-agent Systems,7-14*, 2013.

[40] Young-Ho Kim and Dylan Shell. Distributed robotic sampling of non-homogeneous spatio-temporal fields via recursive geometric sub-division. In *IEEE Int'l Conf. Robotics and Automation, 557-562*, 2014.

[41] Zuoen Wang, Jingxian Wu, Jing Yang, and Hai Lin. Optimal energy efficient level set estimation of spatially-temporally correlated random fields. In *International Conference on Communications*, pages 1–6. IEEE, 2016.

[42] D. Marthaler and Andrea L. Bertozzi. Tracking environmental level sets with autonomous vehicles. *Journal of the Electrochemical Society*, 129:2865, 2003.

[43] R Willett and R Nowak. Minimax Optimal Level Set Estimation. *IEEE Transactions on Image Processing*, 16(12):2965–2979, 2007.

[44] Abhijeet Joshi, Trevor Ashley, Yuan R. Huang, and Andrea L. Bertozzi. Experimental validation of cooperative environmental boundary tracking with on-board sensors. In *2009 American Control Conference*, pages 2630–2635. IEEE, 2009.

[45] David Salda and Renato Assunc. Predicting Environmental Boundary Behaviors with a Mobile Robot. 2016.

[46] Sara Susca, Francesco Bullo, and Sonia Martinez. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16:288–296, 2008.

[47] Dimitar Baronov and John Baillieul. Reactive exploration through following isolines in a potential field. pages 2141–2146. IEEE, 2007.

[48] R Cortez, Xanthi Papageorgiou, H Tanner, A Klimenko, K Borozdin, Ron Lumia, and W Priedhorsky. Smart radiation sensor management. *IEEE Robotics & Automation Magazine, 15(3):85-93*, 2008.

[49] Jinlu Han and YangQuan Chen. Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field. *Jour. Intelligent & Robotic Systems, 323-332*, 2014.

[50] Matteo Reggente and Achim J Lilienthal. Using local wind information for gas distribution mapping in outdoor environments with a mobile robot. In *Sensors, IEEE, 1715-1720*, 2009.

[51] Nannan Cao, Kian Hsiang Low, and John M Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Int'l Conf. on Autonomous agents and multi-agent systems, 7-14*, 2013.

[52] RA Cortez and HG Tanner. Radiation mapping using multiple robots. In *Emergency Preparedness and Response and Robotic and Remote Systems, ANS Int'l Joint Topical Meeting on, 157-159*, 2008.

[53] Jinlu Han and YangQuan Chen. Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field. *Jour. Intelligent & Robotic Systems, 323-332*, 2014.

[54] Cong Wang, Chung-Yen Lin, and Masayoshi Tomizuka. Statistical learning algorithms to compensate slow visual feedback for industrial robots. *ASME Journal Dynamic Systems, Measurement, and Control*, 2015.

[55] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Int'l Conf. on Autonomous agents and multiagent systems, 23-30*, 2008.

[56] Young-Ho Kim and Dylan Shell. Distributed robotic sampling of non-homogeneous spatio-temporal fields via recursive geometric sub-division. In *Robotics and Automation, IEEE Int'l Conf. on, 557-562*, 2014.

[57] Nima Ghods and Miroslav Krstic. Source seeking with very slow or drifting sensors. *Journal of Dynamic Systems, Measurement, and Control, 044504-044508*, 2011.

[58] Shun-ichi Azuma, Mahmut Selman Sakar, and George J Pappas. Stochastic source seeking by mobile robots. *Automatic Control, IEEE Transactions on, 2308-2321*, 2012.

[59] Miloš S Stanković and Dušan M Stipanović. Extremum seeking under stochastic noise and applications to mobile sensors. *Automatica, 1243-1251*, 2010.

[60] Arnab Ghoshal and Dylan Shell. Covering space with simple robots: from chains to random trees. In *Robotics and Automation, IEEE Int'l Conf. on, 914-920*, 2013.

[61] Ajith Gunatilaka, Branko Ristic, and Ralph Gailis. On localisation of a radiological point source. In *Information, Decision and Control, 236-241*, 2007.

[62] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI Conf. Artificial Intelligence, 1650-1654*, 2007.

[63] Kian Hsiang Low, John M Dolan, and Pradeep K Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Int'l Conf. Automated Planning and Scheduling*, 2009.

[64] Howie Choset. Coverage for robotics - A survey of recent results. *Ann. Math. Artif. Intell.*, 31(1-4):113–126, 2001.

[65] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61:1258–1276, 2013.

[66] Derek Mitchell, Nilanjan Chakraborty, Katia Sycara, and Nathan Michael. Multi-Robot Persistent Coverage with stochastic task costs. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3401–3406, 2015.

[67] Grant P. Strimel and Manuela M. Veloso. Coverage planning with finite resources. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2950–2956, 2014.

[68] Ercan U. Acar, Howie Choset, and Ji Yeong Lee. Sensor-based coverage with extended range detectors. *IEEE Transactions on Robotics*, 22:189–198, 2006.

[69] Liam Paull, Mae Seto, and Howard Li. Area coverage planning that accounts for pose uncertainty with an AUV seabed surveying application. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6592–6599, 2014.

[70] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International Journal of Robotics Research*, 22(7-8):441–466, 2003.

[71] Morteza Lahijanian, Matthew R Maly, Dror Fried, Lydia E Kavraki, Hadas Kress-gazit, Senior Member, and Moshe Y Vardi. Environments With Partial Satisfaction Guarantees. 32(3):583–599, 2016.

[72] Sean B Andersson. Curve tracking for rapid imaging in AFM. *IEEE Transactions on Nanobioscience*, 6:354–361, 2007.

[73] C Barat and M J Rendas. Benthic boundary tracking using a profiler sonar. *Iros 2003: Proceedings of the 2003 Ieee/Rsj International Conference on Intelligent Robots and Systems, Vols 1-4*, pages 830–835\r3828, 2003.

[74] S. K. Lee, S. P. Fekete, and J. McLurkin. Structured triangulation in multi-robot systems: Coverage, patrolling, Voronoi partitions, and geodesic centers. *The International Journal of Robotics Research*, pages 1–27, 2016.

[75] O. Arslan and D. E. Koditschek. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *IEEE International Conference on Robotics and Automation*, pages 4259–4266, 2016.

[76] Young Ho Kim and Dylan A. Shell. Distributed robotic sampling of non-homogeneous spatio-temporal fields via recursive geometric sub-division. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 557–562, 2014.

[77] Derek Mitchell, Micah Corah, Nilanjan Chakraborty, Katia Sycara, and Nathan Michael. Multi-robot long-term persistent coverage with fuel constrained robots.

In *IEEE International Conference on Robotics and Automation*, volume 2015-June, pages 1093–1099, 2015.

[78] L Lin and M A Goodrich. Hierarchical Heuristic Search Using a Gaussian Mixture Model for UAV Coverage Planning. *IEEE transactions on cybernetics*, 44:2532–2544, 2014.

[79] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems*, 62(11):1646–1656, 2014.

[80] Matthias Faessler, Flavio Fontana, Christian Forster, Elias Mueggler, Matia Pizzoli, and Davide Scaramuzza. Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics*, 2015.

[81] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart. Versatile distributed pose estimation and sensor self-calibration for an autonomous mav. In *Robotics and Automation, IEEE International Conference on*, pages 31–38, 2012.

[82] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Robotics and Automation, IEEE International Conference on*, pages 20–25, 2011.

[83] Raul Mur-Artal, JMM Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *arXiv preprint arXiv:1502.00956*, 2015.

[84] Luis Rodolfo García Carrillo, Alejandro Enrique Dzul López, Rogelio Lozano, and Claude Pégard. Combining stereo vision and inertial navigation system for a quad-rotor uav. *Journal of Intelligent & Robotic Systems*, 65(1-4):373–387, 2012.

[85] Slawomir Grzonka, Giorgio Grisetti, and Wolfram Burgard. A fully autonomous indoor quadrotor. *Robotics, IEEE Transactions on*, 28(1):90–100, 2012.

[86] Eric Wan, Ronell Van Der Merwe, et al. The unscented kalman filter for nonlinear estimation. In *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158. IEEE, 2000.

[87] L. V. Santana, A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli. A trajectory tracking and 3d positioning controller for the ar.drone quadrotor. In *Unmanned Aircraft Systems, International Conference on*, pages 756–767, 2014.

[88] Thomas C Hales. The honeycomb conjecture. *Discrete & Computational Geometry, 1-22*, 2001.

[89] John Clark and Derek Allan Holton. *A first look at graph theory*. World Scientific, 1991.

[90] Silvano Martello and Paolo Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[91] Donald E McClure and Richard A Vitale. Polygonal approximation of plane convex bodies. *Journal of Mathematical Analysis and Applications*, 51:326–358, 1975.

[92] M. Kemp, A.L. Bertozzi, and D. Marthaler. Multi-UUV perimeter surveillance. In *Autonomous Underwater Vehicles*, pages 102–107. IEEE, 2004.

[93] Dimitris G Tzikas, Aristidis C Likas, and Nickolaos P Galatsanos. The variational approximation for bayesian inference. *Signal Processing Magazine, IEEE, 131-146*, 2008.

[94] Mo Chen. Variational bayesian inference for gaussian mixture model. 2012.

[95] Bo Wang and DM Titterington. Convergence properties of a general algorithm for calculating variational bayesian estimates for a normal mixture model. *Bayesian Analysis, 625-650*, 2006.

[96] John DC Little, Katta G Murty, Dura W Sweeney, and Caroline Karel. An algorithm for the traveling salesman problem. *Operations research*, 11(6):972–989, 1963.

Publications

1. Abdullah Al Redwan Newaz, Sungmoon Jeong, Hosun Lee, Hyejeong Ryu, and Nak Young Chong. "**UAV-based multiple source localization and contour mapping of radiation fields.**" *Robotics and Autonomous Systems* 85 (2016): 12-25.

2. Abdullah Al Redwan Newaz,Sungmoon Jeong, and Nak Young Chong. "**Fast Radioactive Hotspot Localization Using a UAV .**" In 2016 *IEEE SIMPAR*, Accepted. IEEE, 2016.

3. Abdullah Al Redwan Newaz, Sungmoon Jeong, Hosun Lee, Hyejeong Ryu, Nak Young Chong, and Matthew T. Mason. "**Fast radiation mapping and multiple source localization using topographic contour map and incremental density estimation**." *In Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1515-1521. IEEE, 2016.

4. Abdullah Al Redwan Newaz, and Nak Young Chong "**Development of Teleoperated and Semi-Autonomous Aerial Vehicles.**" In *JAIST Repository*, Mar 2014.

5. Abdullah Al Redwan Newaz, Geunho Lee, Ferdian Adi Pratama, and Nak Young Chong. "**3D exploration priority based flocking of UAVs.**" In 2013 *IEEE International Conference on Mechatronics and Automation*, pp. 1534-1539. IEEE, 2013.

6. Abdullah Al Redwan Newaz, Ferdian Adi Pratama, and Nak Young Chong. "**Exploration priority based heuristic approach to uav path planning**." In *2013 IEEE RO-MAN*, pp. 521-526. IEEE, 2013.

7. Abdullah Al Redwan Newaz, Sungmoon Jeong, and Nak Young Chong. "**Boundary Estimation in Partially Observable Environments Using a UAV with Online Manner.**" *Intelligent and Robotic Systems*, Under review.

8. Abdullah Al Redwan Newaz, Sungmoon Jeong, and Nak Young Chong. "**Rapid Regions of Interest Coverage Algorithm based on Adaptive Hierarchical Area Decomposition for an Environmental Monitoring.**" *Intelligent and Robotic Systems*, Under review.