The Thesis Committee for Meredith Leeann Pitsch
Certifies that this is the approved version of the following thesis:

Spatio-temporal Map Maintenance for Extending Autonomy in Long-term Mobile Robotic Tasks

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: Sheldon Landsberger

Co-Supervisor: Mitchell Pryor

Spatio-temporal Map Maintenance for Extending Autonomy in Long-term Mobile Robotic Tasks

by

Meredith Leeann Pitsch

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2018

Acknowledgements

I want to thank God for helping me in this period, and giving me the strength to get through all the tough classes and long nights. I also thank my parents for supporting me, and continuously encouraging me, and my fellow lab members for inspiration, comradery, and fellowship. I especially want to thank my advisors Dr. Mitch Pryor and Dr. Sheldon Landsberger for their invaluable wisdom, direction, and leadership. Finally, I want to thank the University of Texas and Los Alamos National Laboratory for supporting my education both academically and financially.

Abstract

Spatio-temporal Map Maintenance for Extending Autonomy in Long-term Mobile Robotic Tasks

Meredith Leeann Pitsch, M.S.E.

The University of Texas at Austin, 2018

Supervisor: Sheldon Landsberger

Co-Supervisor: Mitchell Pryor

Working in hazardous environments requires routine inspections in order to meet safety standards. Dangerous quantities of nuclear contamination can exist in infinitesimally small volumes. In order to confidently inspect a nuclear environment for radioactive sources, especially those which emit alpha radiation, technicians must carefully maintain detectors at a consistent velocity and distance from a source. Technicians must also take careful records of which areas have been surveyed or not are important so that no area is left unmonitored. This is a difficult, exhausting task when the coverage area is larger than an office space. An autonomous mobile robotic platform with Complete Coverage Path Planning (CCPP) can reduce dangerous exposure to humans and provide better information for Radiological Control Technicians (RCT). The developed robotic system - or RCTbot - is designed for long-term deployment with little human correction, intervention, or maintenance required. To do this, the RCTbot creates a map of the environment, continually updates it based on multiple sensor inputs, and searches its map for contamination. In nuclear environments, the areas of interest often remain spatially constant throughout the duration of an

inspection and are considered temporally static. The RCTbot monitors temporally static environments but adapts to dynamic changes over time. It then uses its sensor data to update and maintain its map so no manual human intervention is necessary. The spatio-temporal map maintenance (STMM) is agnostic to the survey type, so the RCTbot system is viable for application domain other than nuclear.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Robotic systems are rapidly attaining higher levels of autonomy. The Greek roots of the word *autonomous* translates to "self law". The term closely relates to *automatic* which Oxford defines as "working by itself with little or no direct human control." In a robotics sense, autonomy is defined as:

> The extent to which a robot can **sense** its environment, **plan** based on that environment, and **act** upon that environment with the intent of reaching some task-specific goal (either given to or created by the robot) without external control *[1]*.

While most deployed systems demonstrate some autonomy, many fail to reach the strictly limited degree of human intervention required for higher level self-agency [1]. In the last two decades, roboticists and the artificial intelligence community have rapidly advanced the field [2] [3] [4], but gaps in their capabilities remain including how to handle autonomy in real-world environments.

Discussing levels of autonomy requires understanding that the task dictates the degree of self-agency the robot requires [1]. As an example, imagine two robots designed to autonomously complete distinct tasks. The first robot's task is to seek an item and return with it in some finite length of time. The second robot must patrol an area for intruders indefinitely. Theoretically, the patrol robot must retain agency for an infinitely longer time than the seeking robot. Despite this, one cannot argue that the patrol robot is more autonomous than the seeking robot because it is based on each individual robot's purpose. On the other hand, if the patrol robot breaks down during its task and must be repaired by a human actor, it clearly demonstrates less autonomy than the

seeking robot which never required human intervention. This holds true regardless of the duration the patrol robot remained active.

Most robotics researchers implement the task-based autonomy paradigm to some degree, but they often neglect duration of the task as an important factor to model [1]. Deployment in real-world applications requires maintaining autonomy over task-specific lengths of time [5].

Enabling a system to account for changes in its environment is one way to improve its level of autonomy and prepare it for deployment. It is simple to see that the problem of task-specific autonomy is more difficult for indefinite length tasks. This work demonstrates how enhancing a robot's ability to operate indefinitely without human intervention enhances its usefulness in a real-world environment. This chapter specifically will define the environment and task on which the method will be applied and introduce the method and its objectives. A Glossary of terms is provided near the end of this work.

1.1. MOTIVATION

Researchers develop autonomous solutions when removing humans from the system improves outcomes. Historically, tasks that are dirty, dangerous, difficult, or dull attract autonomous systems. As it becomes riskier for humans to intervene, the level of autonomy in a robot must increase; therefore, tasks which are very hazardous typically have greater impetus for autonomy.

The nuclear industrial complex has a variety of tasks and environments that present a biological hazard to human health. The U. S. Nuclear Regulatory Commission (NRC) mandates that a radiation worker should have an annual occupational radiation exposure limit of 5 rem (0.05

Sv) [6]. To mitigate stochastic health hazards at lower level exposures, nuclear industries developed a practice to keep doses to workers *As Low As Reasonably Achievable*, or ALARA [7]. In seeking to practice ALARA, operations in radiologically hazardous environments are limited in exposure duration, carefully monitored to ensure distance from sources, and properly protected by the correct protective equipment [7] [8]. Radiation exposure is a function of time, distance, and shielding [9]. Deploying an autonomous robot for certain radiological tasks removes the worker from potential exposure; however, as systems require more physical human intervention, more of the benefits of deploying a robot are negated. This reality exemplifies the need for higher levels of autonomy in radiologically tasked robots.

1.2. HAZARDS ASSOCIATED WITH NUCLEAR APPLICATIONS

A specific radiological task motivates this effort and will be used to evaluate the effectiveness of the developed approach in a real-world setting. Los Alamos National Laboratory (LANL) performs operations with a number of transuranic radionuclides for the Department of Energy (DOE). The Pit Technologies (PT) group (formerly MET-2) at LANL is specifically charged with a number of operations involving plutonium in various forms. During the course of normal operation, there is a risk of trace amounts of Special Nuclear Material (SNM) escaping containment and contaminating the work area.

The intake of any amount of radioactive plutonium is a potential health hazard [7] [10] as a significant portion of the energy it emits is in the form of alpha particles (with the rest leaving as gamma and neutron radiation). Once in the body, plutonium migrates into various systems depending on the intake path, continuously emitting alpha radiation [11]. This *body burden* is the

amount of received material that is fixed in tissue and eliminated very slowly over a person's lifetime [9]. The body burden continues to release radiation adding to the *committed dose* [10]. A worker with just a single, miniscule exposure to plutonium receives dose from it for the duration of his or her life. Because the effect of any level of exposure increases the long-term risk, facilities take every precaution to ensure workers never intake plutonium.

The PT group completes all plutonium related tasks at the LANL Plutonium Facility (PF-4). To maximize worker safety, the facility has a system of controlling and containing radiation and contamination. PF-4 has different zones based on relative hazards [8]. LANL recognizes Radiological Controlled Areas (RCA), Radiological Buffer Areas (RBA), and Radiological Areas (RA) as those which are primarily associated with potential exposure risk according to DOE guidelines for plutonium facilities [7] [8] [12]. In PF-4, plutonium processes are completed in RAs with gloveboxes as shown in Figure 1-1.

Figure 1-1: A worker demonstrating safe processes in a glovebox [13].

Gloveboxes are designed both to keep workers safe and to optimize the processes. The environment is pressurized such that the negative atmosphere keeps radioactive particles inside away from the workers. The tightly controlled RA also limits elements damaging to processes such as oxygen which could react with the plutonium. Outside of the gloveboxes, a RBA controls for any small escape of contamination due to a breach in the containment. It is a second layer of defense to prevent the spread of any contamination outside of the RCA which encompasses all areas where workers should expect possible radiation levels.

1.2.1. Radiological Survey Task

As a means of verifying the integrity of the containment and control system, DOE standards [7] [12] recommend surveying all areas that could potentially become contaminated, or the entire

RCA. LANL has adopted these standards and implemented a program for radiation protection. The objectives of radiation monitoring found within the program's documentation [8] are listed as follows:

a) Demonstrate regulatory compliance;
b) Document radiological conditions;
c) Characterize workplace conditions and detect changes in those conditions;
d) Detect the gradual buildup of radioactive material in the workplace;
e) Verify the effectiveness of engineered and administrative controls in containing radioactive material and reducing radiation exposure;
f) Identify and control potential sources of personnel exposure; and
g) Determine exposure rates as necessary during each entry to a High or Very High Radiation Area (HRA or VHRA)

Monitoring radiation and contamination levels both instantaneously and over time is a key component of the objectives.

A worker responsible for radiation monitoring is known as a Radiological Control Technician (RCT) [7] [8]. Contamination surveys comprise a small portion of the responsibilities of a RCT, but the time spent on the task is dependent on the size of the area surveyed and the monitoring frequency. Long surveys remove RCTs from other duties including maintaining and improving protective containment and controls. In the case of a RA with non-negligible dose rates, RCTs are exposed to the hazard for longer periods of time. Despite the possible danger of long surveys, it is necessary to fully characterize the radiation and contamination within RCAs and minimize worker dose.

1.2.2. RCTbot

To address the hazard to RCTs and improve the efficiency of their tasks, PT has proposed using a mobile platform to complete the task known as *RCTbot*. The first generation of RCTbot is shown in Figure 1-2. The RCTbot serves as a tool for RCTs to survey RCAs remotely. Ideally, the system monitors large areas autonomously for long periods of time, requiring RCT intervention only when unexpected contamination is found. The system must navigate RCAs, take radiation measurements, alert an RCT to detected radiation, and maintain a spatially and temporally accurate record of the entire space. The robot will need to perform better – safer, longer, more reliably – than a human in order to meet the criteria for dose limitation practices in a DOE plutonium facility. The guidelines state that "no practice shall be adopted unless its introduction produces a positive net benefit [7]" which is to say that a robot must quantifiably improve worker safety in order to be deployed.

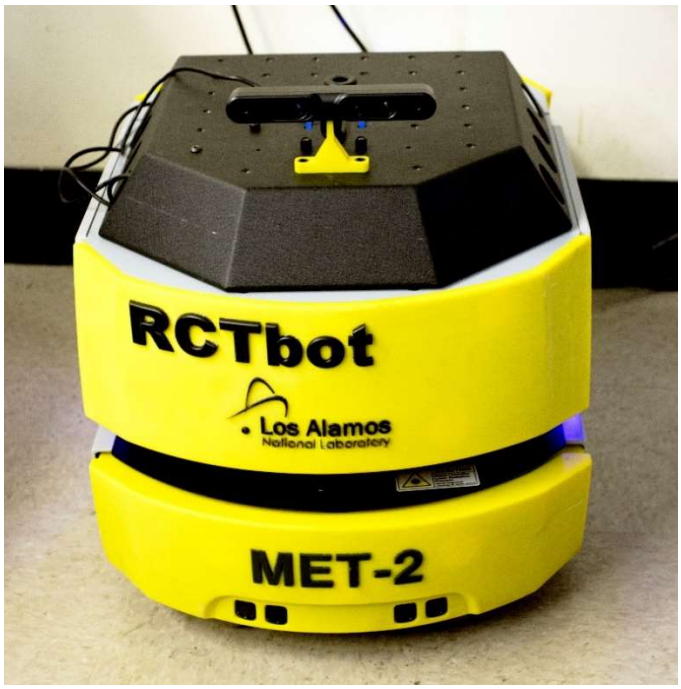

Figure 1-2: First generation implementation of RCTbot.

Thus, before the system is deployed, PT must prove that the robot will produce a net benefit. If the robot is able to consistently survey areas over time without RCT intervention, the RCTs can better focus their efforts on increasing overall worker safety as well as reducing their own exposure time in case of contamination. Furthermore, RCTbot offers a means to collect better data on the surveys than is possible using human operators. In order to navigate the RCA and keep the required record of contamination, the RCTbot continuously updates both its spatial and radiation maps. We propose that the benefit to RCTs increases with the level of autonomy of the RCTbot. The technical challenges that must be addressed are listed in the next section.

1.3. TECHNICAL CHALLENGES FOR MOBILE SYSTEMS

The radiation survey can be broken down into more primitive robot jobs. RCTbot must ideally 1) navigate the area, 2) avoid collisions with static and dynamic obstacles, 3) measure for radiation at specified locations in the RCA, 4) alert RCTs to detected contamination, 5) communicate its results to supervising operators in a clear and archivable format, and 6) complete the above tasks reliably and routinely. Each of these more primitive jobs must be considered for the systems to demonstrate the required level of autonomy for deployment in PF-4.

1.3.1. RCTbot Navigation and Collision Avoidance

Autonomous navigation has been a well-studied area in mobile robotics [14] [15]. Most navigation schemes involve using various sensors to create a map of the environment. When the robot later matches its sensor data to the map, it can estimate its position using probabilistic techniques [14]. Simultaneous Localization and Mapping (SLAM) is a common technique in navigation in which the robot explores an unknown environment and builds a map as it goes, constantly keeping track of its location relative to what it has previously mapped [14]. Autonomous navigation requires the robot to generate a map and localize within it without a human in the loop. The current RCTbot platform performs autonomous navigation by using SLAM to generate maps of the RCAs, and thereafter implementing Adaptive Monte Carlo Localization (AMCL) [14] to track its position or localize. This and many other techniques for completing navigation and collision detection exist. Thus, the focus of this effort is their implementation instead of their development.

A subset of navigation required for RCTbot is Complete Coverage Path Planning (CCPP). In this type of robot navigation an autonomous system generates a plan to visit every accessible point in the environment [15] [16]. Complete and verifiable coverage is required for RCTbot to guarantee that an area is contamination free. Numerous CCPP algorithms exist and will be discussed in detail in Chapter 2. Chapter 3 will compare algorithms and discuss implementation for the survey task.

1.3.2. Radiation Measurements

The survey of the entire RCA requires both coverage planning and sensor integration. RCTs currently use various methods to survey different parts of the RCA which will be discussed further in Chapter 3 [17]. In one configuration, the current RCTbot platform monitors the entire floor. In another configuration, the system may move to particular locations (designated by the operator or in a grid) to take readings. The survey requires the correct integration of radiation detection sensors. LANL uses alpha sensors for the detection of SNM like plutonium which have very specific requirements for valid surveying. The type of sensor is an important consideration for both robotic and radiological reasons. Examples of an alpha detection meter and probe used at LANL are shown in Figure 1-4 and Figure 1-3.

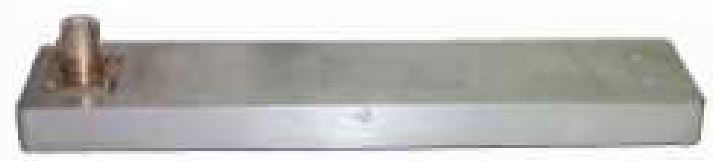

Figure 1-3: Ludlum Model 43-32 Air Proportional Probe [18]

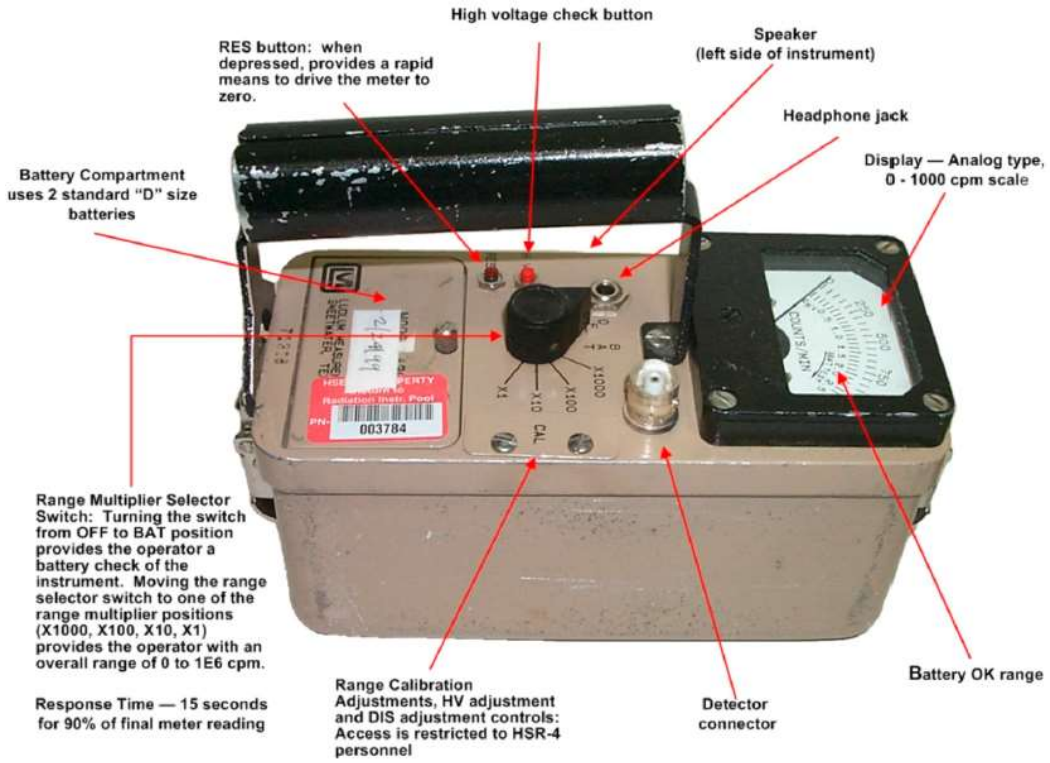

Figure 1-4: Ludlum Model 139 Radiation Survey Meter [18]

Most coverage algorithms assume that if the cross-sectional area of the robot passes over a point, it is covered. In the case of the radiation survey, algorithms need to be adapted such that the area considered corresponds to the active surface of the detector probe.

To complicate navigation matters, the detection of radiation is inherently probabilistic because "radioactive decay is a random process [19]." Uncertainties in both the measurements and detectors require longer counts in order to adequately quantify radiation. The detectors become more useful for sensing low levels of radiation, the longer they are in range of a radiation source. This means that the chance of a moving detector sensing radiation improves given greater probe surface area, smaller source-to-probe distance, and slower navigation speeds. By design, a robotic platform maintains constant detector position and velocity more reliably than a human operator.

Humans are prone to distraction since finding contamination is rare, ergonomic concerns, and the mundanity of the task. In addition, the use of robotics, allows RCTs to rethink what kind of sensors and surveys are possible. The impact of these new design opportunities will further drive the navigation requirements and are discussed in Chapter 3.

1.3.3. Contamination Detection Management

If PT's control and containment systems never fail, the RCTbot will continually survey and find no radiation or contamination. In the real-world, statistics and experience show that the robot will eventually encounter contamination. As a minimum requirement for deployment, the robot must not spread contamination and make the problem worse. How the RCTbot handles positive readings may change over time, but initially, the robot should halt its survey path and signal an RCT for assistance. The way in which RCT is contacted is a subject more suited for a Human-Robot Interaction study, so for the scope of this work, the robot will turn on warning lights, sound an alarm, and note the location of the contamination on its survey map.

Once an RCT has handled the contamination, the RCTbot must decide how to proceed. A number of compromising events could have happened that require a decision from the robot. Depending on the length of time the robot was halted, it may be prudent to start a completely new survey. It is important that the robot handles the partial survey from before if it is unable to complete it in a timely manner. The robot might also have to deal with a depleted battery, causing it to find a charging station, recharge, and then remember where it left off in the survey. In the future, the RCTbot might be required to assist the RCT with the cleanup, necessitating much more

complex task planning capabilities. Chapter 3 gives details on how RCTbot is designed to handle positive readings for the purpose of this work.

1.3.4. Robust and Routine Operation

Survey tasks must be completed robustly and routinely. Real-world environments demonstrate spatio-temporal characteristics. The area's physical layout changes over time in ways that can disrupt normal robot operation. Robust operation at future times and in mutable environments must be addressed. RCTs are responsible for surveying the RCA from initial operation to deactivation and until final Decontamination and Decommissioning (D&D) [7]. Thus, the survey task is effectively indefinite. For a robot, this sort of long-term task is challenging given the nature of the task will inevitably and significantly change over its tenure. It requires the robot retain autonomous behavior in terms of navigation, sensing, and record keeping.

Long-term autonomous navigation in mobile robotics has not been adequately addressed in the literature. Robots navigate by collecting sensor information and building a static map of the environment. As it moves through the environment, a robot will match sensor information with the known map and ensure its position is defined. It is generally assumed the map remains true to the environment for during the assigned task, and that anomalies are temporary and can be addressed and a case-by-case basis and then forgotten. Currently, there is little need to remember the robot encountered an operator in the hallway at a particular location on a particular night as this information is not useful for future planning. What matters is the robot has the capacity to plan for the task in the static environment where the task (i.e. survey) is completed. The most popular method for long-term navigation is to use continuous SLAM, discarding past memory of areas as

new information is received. Continuous SLAM is memory intensive, but modern processors are able to handle the requirements. As a result, many deployed, autonomous robots demonstrate some form of this technique.

Coverage algorithms only guarantee completeness for fully-known maps. Some methods using continuous SLAM are used for exploration and coverage of all known spaces, but antagonistic examples can always be found to show incomplete coverage. If the coverage problem is vacuuming a floor in a house, this is adequate. Incomplete coverage is unacceptable when radioactive contamination could exist in a traversable area and be missed by the RCTbot. A complete map of the environment is required. In real-world applications such as the busy production areas of PF-4, the environment will change over time. This change will be reflected in the robot's sensor measurements, but not the static map. The static map can no longer be discarded as in continuous SLAM because it contains information that the robot's sensors might not be able to find without it. Still, the map cannot remain unchanged as the only reference point for the robot to localize. The discrepancies in the static map will eventually accumulate to the point where the robot cannot reliably complete the survey. To avoid this, the static map must be updated to match the real, spatio-temporal environment. Thus, it is necessary for the navigation package to determine the difference between changes in the static environment and dynamic changes that may only exist for one or few runs.

A robot must be able to maintain its static map over time. This Spatio-Temporal Map Maintenance (STMM) is necessary for long-term autonomy. Not only must the robot retain the ability to navigate in the space, but it must also produce accurate, interpretable contamination maps for RCTs. If an RCT cannot recognize places of importance on the robot's map, the map is useless.

The RCT will have to manually survey the real environment, and the robot has failed to remove the necessary human intervention. Chapter 4 of this work discusses how a robot uses STMM to autonomously update its belief about the environment over time in order to increase its usefulness when deployed in a real-world environment.

1.4. OBJECTIVES AND ORGANIZATION

The sections above describe problems in deploying autonomous systems into real-world environments. The length and routine nature of a task are often neglected when new autonomy algorithms are presented in literature. Robots incapable of maintaining autonomy over time negate their usefulness and human operators will remove them from the task. This thesis seeks to address some of the barriers that autonomous robots encounter with long-term deployment.

The main objective of this work is to demonstrate the readiness of an autonomous robotic system for deployment. For the RCTbot task described previously, the system must continually, and autonomously perform a radiation survey for an indefinite length of time. The robot must also prove itself discernably better for the task than a human RCT. To accomplish these goals, the robot should be capable of high-fidelity navigation through large spaces, accurate radiation sensing, verifiably complete coverage, competent planning and task management, and self-updating its environmental beliefs.

If the RCTbot system can demonstrate the capabilities listed above, it will be ready for deployment into the real-world environment of PF-4. The following chapters will show how this is accomplished:

- **Chapter 2** provides background on how robots have performed in situations similar to the RCTbot task. It gives examples of popular navigation algorithms including continuous SLAM and complete coverage. Finally, it discusses progress in the literature toward spatio-temporal updating of robot maps.
- **Chapter 3** describes how humans handle the radiation survey task and proposes the methods to be used by the RCTbot. It compares various coverage algorithms and discusses which is implemented by the RCTbot. Finally, it shows the verification process for guaranteeing complete coverage. Special attention in this chapter is given to a single cycle of the survey task.
- **Chapter 4** extends the work described in Chapter 3 to completing the task routinely over an extended period of time. It introduces the RCTbot's STMM method for updating beliefs about a slowly changing, spatio-temporal environment and explains how these updates influence decision-making in the system.
- **Chapter 5** sets up an experiment to prove the validity of the proposed method. It parameterizes the task and lists metrics for completion including data, results, and discussion of the experiment are also provided in this chapter. A comparative analysis summarizing the anticipated benefits relative to manual survey methods will be discussed.
- **Chapter 6** summarizes the findings of this thesis. It discusses the implications of the work and makes suggestions toward future research areas.

Chapter 2: Literature Review

Researchers have previously approached various aspects of the radiation coverage problem. This chapter breaks down the principle technical challenges and discusses how others have addressed them in past works. The first section outlines how the radiation survey problem is handled currently with and without autonomous systems. Past attempts to incorporate robots into nuclear work inform how new methods can be applied to the problem at present. After the brief discussion of nuclear topics, the next section presents an overview of the robot coverage problem and how researchers have developed it over time. The final section presents all the current research into spatio-temporal updating as it relates to robotic localization and mapping.

2.1. Radiation and Robotics

As in any hazardous industrialized task, rigid guidelines for nuclear work attempt to maximize personnel safety. Any autonomous system that is to accomplish the same task must meet or exceed the current standards. These standards stem from decades of research and studies with constant improvement and feedback from nuclear facilities [20] [21]. In the nuclear industry, modern guidelines all relate to the ALARA principle [21] [22]. The guidelines in literature seek to provide solutions to some of the greater difficulties encountered in meeting federal requirements.

Most of the difficulties associated with radiation contamination surveys involve handling the probabilistic nature of radiation. Several texts describe methods for handling different radiation measurements [19] [23]. In the case of alpha contamination, studies have determined the appropriate source-to-detector distance, scan speed, and minimum detectable activity (MDA) to

ensure that detectors find any radiation source that could be hazardous to humans during a survey [12] [21].

One of the main challenges of robotic surveying is operating within the appropriate restrictions required to optimize radiation detection. Previous work shows that robots are capable of radiation surveying. Klimenko *et al.* demonstrate how weak radiation sources can be found efficiently in a wide search area via a sequential search [24]. In the sequential search, a technician moves a radiation detector from sector to sector of an area. The measurement duration is shorter for sectors that are likely to be clear of contamination. Compared to a uniform search where each sector is given the same attention, the sequential search decreases the time needed to find sources. Kumar *et al.* extends this work by automating the search with robots [25]. Finally, Cortez *et al*. designed and implemented a real system as shown in Figure 2-1 to perform a fast search for radiation sources after an emergency using the described sequential search method [26].

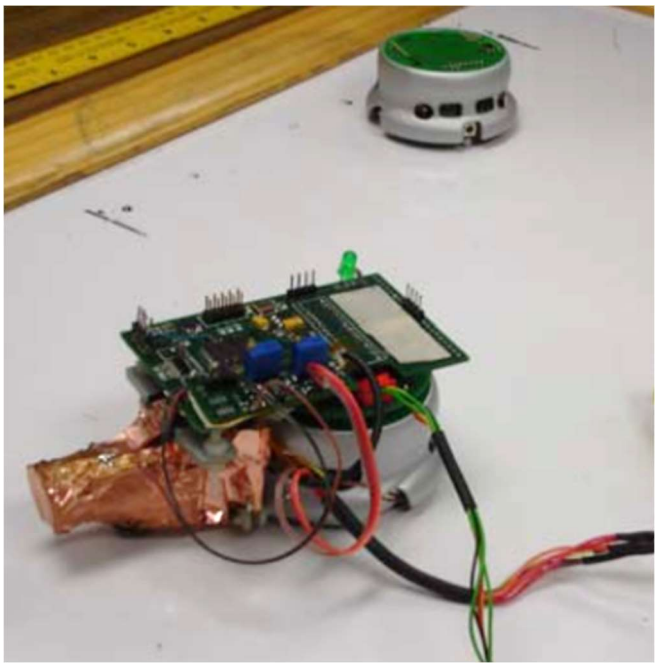

Figure 2-1: Khepera II mobile system with installed radiation sensor [26].

The progression of the works listed above results in an autonomous system for a similar radiation detection problem with a very different scope. The Khepera system is designed to find a source as quickly as possible in a large space during an emergency response situation. Conversely, the RCTbot will handle routine, long-term radiation detection where the optimization problem is based on certainty and not speed.

In addition to the system described above, robotics has been used in other nuclear applications. Anderson and Ebersole both present a review of the history of robotics and automation in the nuclear field [27] [28]. The applications included master-slave teleoperation for remote manipulation in hot cells, systems for work in nuclear power plants, disaster exploration robots such as the ones used at Three Mile Island, Chernobyl, and most recently Fukushima Daiishi, and efforts at Idaho and Savannah River National Laboratories to use mobile teleoperated systems for radiation inspection. Most systems deployed in nuclear environments in the past have been teleoperated such as the national lab robots or purely exploratory like the disaster evaluation robots. Little work has been done to create an autonomous system that learns the environment and executes a routine task for long periods of time.

2.2. COMPLETE COVERAGE PATH PLANNING

Researchers have studied autonomous navigation planning since the first mobile platforms. In the early-1990s, researchers began extending traditional point A to B planning methods into coverage path planning (CPP) [29]. Early attempts started with proven navigation methods such as graph-based searches and the travelling salesman problem [30] [31] [32] or utilized a heuristic approach for long-term probabilistic coverage [33] [34].

In 2001, Choset wrote an extensive review of CPP algorithms up to that point and solidified the terminology used by most researchers [16]. Choset had two major contributions: a categorization of approaches based on how they decompose the area and a demonstration of how optimization parameters drive the choice of algorithm. Robot environment division schemes fall into one of three categories: approximate, semi-approximate, and exact cellular decomposition. Researchers first coined these terms for traditional robotic path planning, but Choset's use standardized them in CPP literature [35]. Choset shows how optimization criteria drive the choice of algorithm. Some tasks such as Cortez's disaster response Khepera system [26] require optimization of speed, while others require optimization of total time [36], overlapping regions [37] [38], fuel economy [39], and other criteria [40] [39] [41]. The different approaches will be discussed further in sections 2.2.1-2.2.2, and Chapter 3 discusses how optimization drives the algorithm choice for this effort.

Continuing Choset's work, Galceran updated the CPP review in 2013 and attempted to systematically categorize past efforts based on more recent trends in the research [15]. The importance of optimization criteria led to implementations that were beyond classical decomposition methods [38]. Galceran instead uses a different aspect of classification that Choset introduced where planning is either done *online*—as the robot moves through the environment—or *offline*—before the robot begins its task [16]. This dichotomy is used throughout this thesis to guide the discussion and compare planning methods. The following sections provide definitions and examples of each category. It must be noted that the following sections provide sufficient examples for the reader to understand the principles of different coverage algorithms and not a full listing of all algorithms developed. For an exhaustive list, the reader should consult [15].

2.2.1. Online Coverage Systems

Online planning algorithms are akin to "sensor-based coverage" because they rely on data from robot sensors instead of a known model of the world [16]. The robot generates coverage plans as it moves around its environment and discovers more territory. Because sensor-based methods are often reactive by necessity, researchers use them for dynamic environments where it is difficult to predict the state of the world at any given time. Galceran and Carreras present many examples of online coverage in a summary of CCPP techniques. The majority follow the same basic principles of discovering cells during exploration and covering them one by one. The following section gives examples of online coverage planning and explains the common principles applied.

The first example of online planning is Morse Decomposition [42]. In Morse Decomposition, a vertical slice is projected across a map and inflection points define cell boundaries as shown in Figure 2-2. This is a form of exact cellular decomposition.

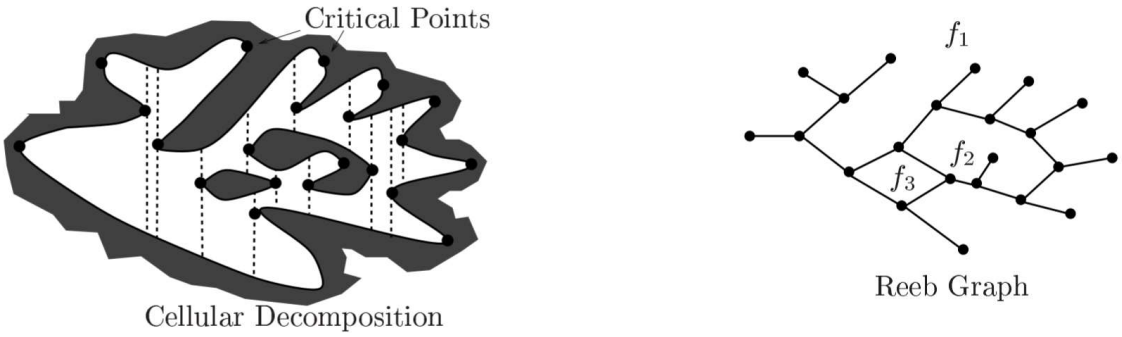

Figure 2-2: Morse decomposition with Reeb Graphs [43]

The critical points form a Reeb graph which yields a navigation plan from cell to cell. The robot travels the graph edges depth-first to examine all uncovered cells and covering the area within using the *boustrophedon* method shown in Figure 2-3 or similar coverage paths [16].

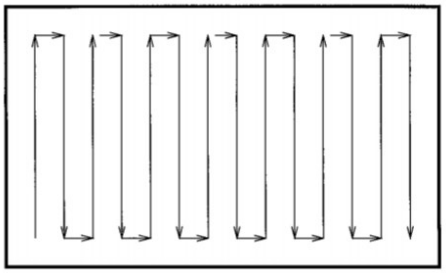

Figure 2-3: Boustrophedon path [16]

Traditional Morse Decomposition requires a known map for offline computation [42]. Acar *et al.* extended the method to online problems by combining a loop closure algorithm based on Euler's formula with the Reeb graph [44]. In this method, the robot can only identify critical points in its line of sight, therefore, a wall-following path with retracing is necessary for completion. The wall-following path is done for discovery of critical points to add to a dynamic Reeb graph followed by the boustrophedon path for coverage in the encircled cell. Given an environment with little change during a single coverage attempt, Acar proves that online Morse Decomposition can perform complete coverage by travelling through the dynamic Reeb graph.

Similar to Morse Decomposition, Landmark based decomposition involves sweeping a line across an area to divide it into cells [45]. Instead of critical points, a number of landmarks are used for simpler classification. The robot treats these geometric obstacles as events and creates a new cell at each event. The event classifications are shown in Figure 2-4.

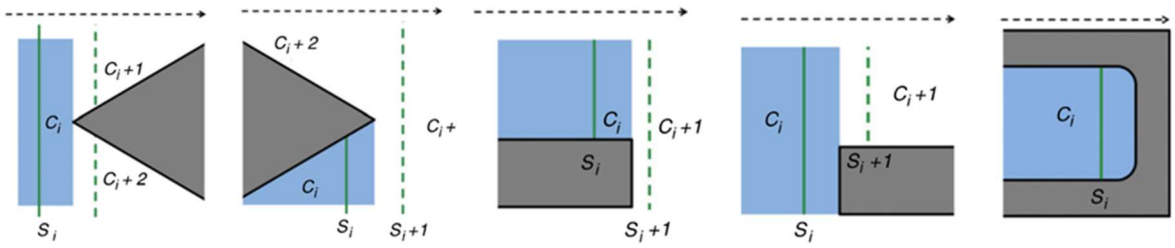

Figure 2-4: Topological landmark events [15].

Each event forms a node in a planar graph representing a topological map of the environment. The edges in the graph represent the method of travel between nodes and give information about the distances between landmarks [45].

Furthering this method, Wong developed an online implementation of Landmark coverage based on a Finite State Machine (FSM) with three states [46]. A model of the FSM is shown in Figure 2-5.

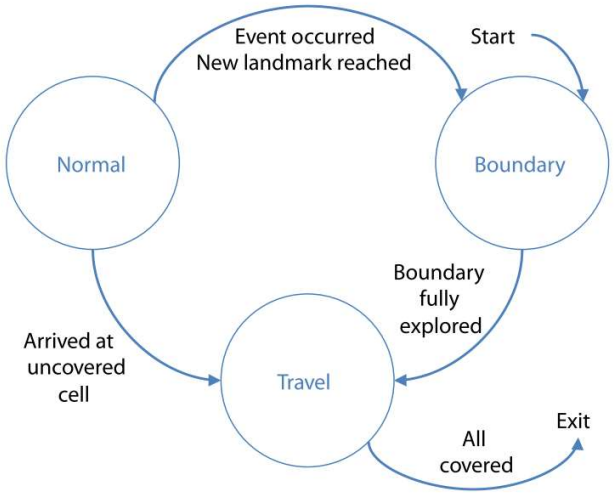

Figure 2-5: FSM describing how to cover an area based on landmark boundaries [46].

In the FSM, a robot begins in a *boundary* state such as a corner. While in the boundary state, it explores with the goal to identify all the surrounding cells and add them to the dynamic topological map. When the robot has completely explored the border of the cell, the system transitions into a *travel* state where it drives to an uncovered cell in the topological map. On arriving, the robot again transitions to a *normal* state. In the normal state, the robot covers the current cell. Once it completes coverage, the system transitions into the travel state again. If in the process of travelling or

covering, a new event occurs, the FSM moves to the boundary state to explore more. In this manner, the entire space is eventually covered.

Other types of coverage are often based on slight differences in decomposition techniques. The principles remain consistent with those examples given above. Genetic algorithms and neural nets use approximate decomposition for online planning. Robots implementing neural nets discretize the space into a grid of neurons as shown in Figure 2-6.

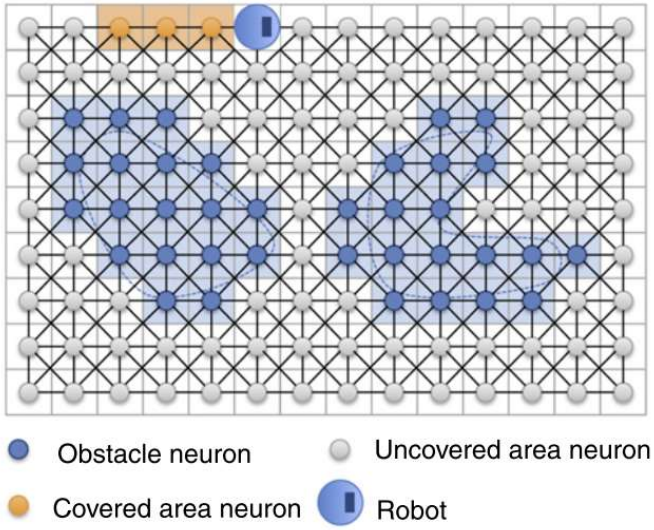

Figure 2-6: Neural network coverage [15].

Each neuron contains various information about the area it represents including if it is an obstacle or whether or not the robot has covered it [47]. This information is used to optimize the coverage time by driving the robot towards areas of lower coverage concentration [48].

All online methods attempt to optimize coverage of an unknown area with respect to completeness using reactive behavior. It is important to note that while the environment does not need to be known in advance, the majority of online methods require the environment to remain static through the duration of the coverage task. If some part of the environment changes in a way

that prevents the robot from detecting a region, that region will not be covered. Furthermore, the robot will not know that the area was not covered and will report completeness. Chapter 3 will discuss the repercussions of this problem in further detail.

2.2.2. Offline Coverage Systems

Offline planning algorithms depend on reliable *a priori* information about the world. They assume a static model of the world in order to generate a full coverage path before the robot begins motion. Robots implementing offline algorithms use sensors only to ensure that the pre-planned path is being followed. Pure offline systems do not react well to changes in the environment as they require matching sensor data to the static map to remain localized. The section below provides examples of offline coverage implementations and describes the general principles applied.

The first example of offline coverage algorithm is wavefront propagation [30]. In wavefront propagation, the robot requires a known map of the environment. It discretizes the map into equal-sized cells using Choset's approximate cellular decomposition classification and then differentiates the obstacle cells from the open cells. Planning a path is done in several stages. First, the robot chooses a start cell and a goal cell. Beginning from the goal cell, each adjacent cell is labelled 1, each cell adjacent to a 1 is labelled 2, and so forth until the entire grid is covered by the propagating wave. An example is shown in Figure 2-7.

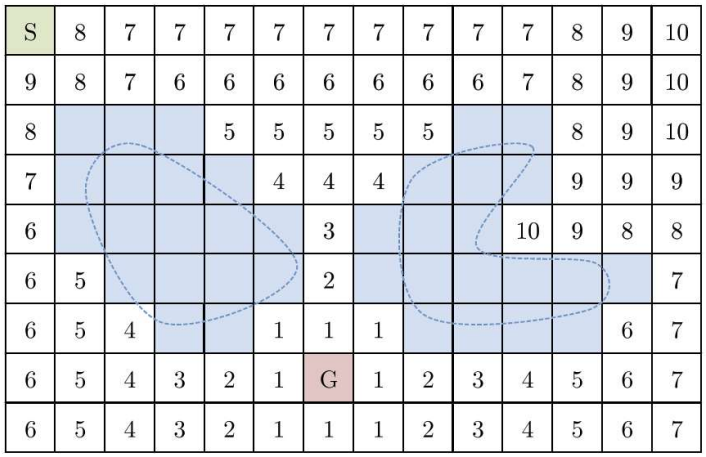

Figure 2-7:Wavefront propagation for coverage [15].

The blue cells in the figure are obstacles, so the robot does not consider them when propagating from cell to cell. Obstacles cause the wave to move around it, so certain cells that are closer in direct distance to the goal have higher labels according to the driving distance the robot must take.

Traditionally, wavefront propagation is used to go from a start point to a goal point via gradient descent. The robot moves from the start point to the next adjacent cell with the lowest numerical label. This guarantees the shortest path between two points, and it is widely accepted as an optimal execution strategy for point to point navigation. Adapting wavefront propagation to coverage navigation requires a simple adjustment from gradient descent to ascent. From the start point, move to the next cell with the highest numerical label. In this way, the robot will cover every cell in the space until the only remaining cell is the goal point with a label of 0. Figure 2-8 shows this applied to the map shown in Figure 2-7.

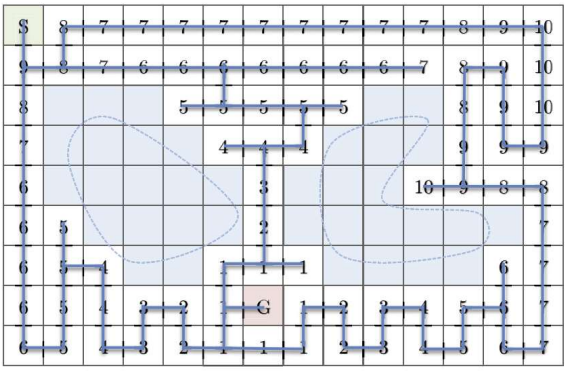

Figure 2-8: Path generated by a wavefront planner [15].

When the plan path first reaches the 1 labels around the goal, it follows the gradient of 1's until it sees a 2 label. From that point, it explores that path until it eventually returns back to the 1's around the goal. This form of coverage guarantees coverage of the space with some retracing of paths.

The next example of offline planning is the Spiral Spanning Tree Coverage (STC) algorithm designed to eliminate retracing [49]. Like in wavefront propagation, the algorithm requires a known, static map to decompose into cells. Dissimilarly, the resolution of the cells must be four times the scan cell size so each mega cell is made of four coverable quadrants as shown in Figure 2-9.

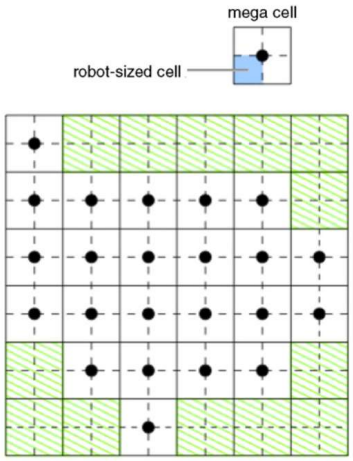

Figure 2-9: Grid for the Spiral-STC method [15].

This form of discretization is rougher than other types, and so only simpler, rectilinear environments can be covered.

To cover the space, the robot plans to reach each mega cell. When traversing the mega cells, it follows around the minor cells on the outside boundary of the entire space. When the robot covers part of each mega cell, it retraces through the minor cells until it reaches the start again in the spiral shape from which its name derives. Figure 2-10 shows the coverage path.

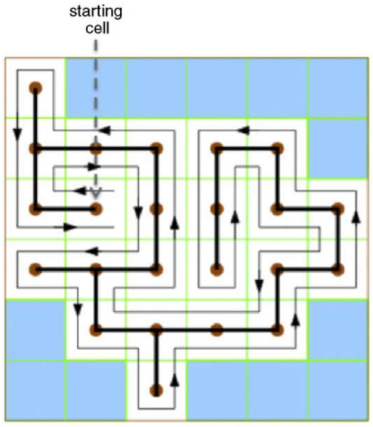

Figure 2-10: Path generated by the Spiral-STC method [15].

The robot must traverse all of the minor cells in a mega cell when they first enter it if it is only connected to the graph by one edge. Otherwise the robot only partially covers the mega cell and finishes it when travelling back to start.

Using the Spiral STC method, a robot is sure to find an optimal path through the space without retracing. It is limited by the resolution of the mega cells in the map however, and not very useful for more realistic environments.

2.3. SPATIO-TEMPORAL SYSTEMS

In contrast with coverage planning, spatio-temporal research is a relatively new topic for robotic systems. Foundational work began with Biswas in 2002 [50], leading to multiple documented studies being published in 2005 [51] [52] [53] [54]. Many papers focus on localization improvements [53] [55] [56]. Some later papers examine spatio-temporal insights into patterns [57] [58]. Others focus on improving general mapping techniques [59].

The types of sensors used for determining elements of spatio-temporal characteristics also vary from paper to paper. Some focus on stereo images [55]. Others look at range data from LIDAR units [56] [60]. A group of researchers from MIT has developed an abstract framework for generic feature-based mapping [59], which is the basis for work in Chapter 4.

While this effort shares many of the same ideas present in literature, the implementation details required for the real-world environment described in Chapter 1 differ. Chapter 4 provides more information about the application of spatio-temporal reasoning in real-world environments.

2.4. SUMMARY

A review of recent efforts in radiation surveying, robotic coverage planning, and autonomous map maintenance showed that additional efforts are necessary to accomplish autonomous radiation coverage surveys over long periods of time, where it will not be feasible to manually re-map the areas for every change in the environment.

Each section above describes different aspects of the overall problem and discusses how researchers have approached similar problems in the past. Section 2.1 describes previous work in radiation surveys. While previous research focused on quickly finding a radiation source known to exist, no one has addressed the problems of searching for low levels of contamination that may

not necessarily be present. This effort shows how a robot can reliably survey for incidental radiation on the ground in large spaces. Next, Section 2.2 identifies multiple coverage algorithms in the established literature that could be implemented. The detailed descriptions support the selection methodology Chapter 3 presents for the application to radiation surveying. Finally, Section 2.3 introduces concepts from spatio-temporal reasoning research. The previous efforts to address the spatial-temporal issues each succeed in demonstrating increased autonomy in certain situations, but most do not meet several requirements identified for this task including:

- Map maintenance
- Map prediction
- Map legibility

To address these issues, this effort proposes a STMM method based on work presented in [59]. Chapter 4 provides explanations of the theory of the STMM algorithm and Chapter 5 provides an experiment and evidence for its effectiveness.

Chapter 3: Alpha Contamination Survey

Before it is deployed in real environments, the RCTbot must accomplish the radiation survey task at least as well as current methods, while improving in areas such as worker safety, survey accuracy, and scan frequency. This chapter provides a detailed description of how the RCTbot completes an alpha contamination survey. It includes a description of the task requirements, current RCT methods, a comparison of CCPP algorithms, a description of chosen algorithm and implementation details, and explanation of the verification process.

3.1. RCT SURVEY

The Code of Federal Regulations (CFR) includes requirements for "Monitoring of Individuals and Areas" as it relates to 10 CFR 835: Occupational Radiation Protection [61]. From these requirements, the DOE created recommendations of controls for radiological tasks [12] and specifically adapted these for plutonium facilities like the one at LANL [7]

LANL must abide by the regulations put forward in 10 CFR 835 as part of its federal contract [8]. The lab meets the requirements by implementing procedures developed according to the DOE guidelines [7]. The specific procedures for the survey task designated in Chapter 1 are outlined and described in a set of internal work and process control documents [17] [62] [63] [64].

3.1.1. Requirements

A ground radiation contamination survey is only admissible under certain criteria. Many of the criteria outlined in the process control documents do not relate to autonomous surveys. For example, insuring proper instrument calibration and RCA designation will still be the

responsibility of RCTs. Specifications require that a surface survey is admissible if and only if the detector is maintained at a distance no greater than 0.25in (0.5cm) from the surface and moved at a velocity no greater than 2 in/s (5 cm/s) [62]. These requirements are deemed necessary to ensure the detection of contamination above designated radioactivity control levels [7] [8]. Since a robot may allow more precise tracking, the documented requirements may be more conservative than necessary to accommodate imprecise humans. Section 3.1.2 briefly discusses the requirements and the velocity requirement is reassessed using radiation detection theory.

3.1.2. Radiation Detection Theory

The following sections describe the theory and methods used in radiation detection. The first section presents terminology and definitions for radiation counting. After that follows an explanation of error handling. Finally, the last section shows how the detector velocity requirement can be adjusted for better compliance and safety. The theory presented here is consistent for both human and robot surveyors alike.

3.1.2.1. Radiation Counting

Consider a radioactive point source that decays at a constant rate. The number of decays or disintegrations per second (dps) is defined as the activity, A. A detector measures activity by counting the number of energetic interactions per unit time in counts per minute (cpm) or counts per second (cps). In most cases, the true count rate from the source is intermingled with background noise. Typically, alpha radiation detectors are not affected by background due to the limited distance alpha particles can travel in air, but for completeness, background will be included in the following theory.

The background count is removed from the gross count by

$$r = g + b \ [cpm]$$

$$g = \frac{G}{t_G} \tag{1}$$

$$b = \frac{B}{t_B}$$

where r is the total count rate and G and B are the gross and background count respectively taken over a duration t_G and t_B. The best estimate of a mean for a single count G or B is the standard deviation of the count [23]. The standard deviation applies to the gross and background counts as

$$\sigma_G = \sqrt{G}$$

$$\sigma_B = \sqrt{B} \tag{2}$$

From [19], it is shown that the uncertainty in the difference of counts is the sum of their variances. Using (1), this is shown by

$$\sigma_r^2 = \sigma_g^2 + \sigma_b^2$$

$$\sigma_r = \sqrt{\sigma_g^2 + \sigma_b^2}. \tag{3}$$

Substituting (2) into (3) transforms it into

$$\sigma_r = \sqrt{\frac{G}{t_G^2} + \frac{B}{t_B^2}} \tag{4}$$

with the assumption that the variance of time is negligible. Using (1) and (4), the resulting count rate is written

$$CR = r \pm \sigma_r \tag{5}$$

where is the raw number typically recorded in surveys. Standard error propagation can be used to determine the error for the remaining calculations, but it is omitted here for brevity.

3.1.2.2. Expected vs. Experimental Counts

In order to obtain the activity from the count rate, the application of a correction equation adjusts the activity. The correction equation is

$$A = \frac{CR}{\epsilon APB} \tag{6}$$

where CR is from (5), ϵ is the detector efficiency, A is the self-absorption factor, P relates to sample preparation in testing environments and can be ignored, and B is the backscatter factor. Because the robot only seeks to detect those particles that exist at the detector, both self-absorption and backscatter can be ignored. These are both necessary to determine the true activity, but the dangerous activity, in this context, is only that which is detectable. Efficiency is divided into two types, intrinsic and absolute. Intrinsic efficiency is the detector efficiency measuring the number of pulses incident on the detector over the number counted. Absolute efficiency is the number of pulses emitted by the source over the number detected. Absolute efficiency is a function of the solid angle between the source and detector. For alpha radiation, the angle is less important than the distance between them. Given this, the activity in (6) becomes

$$A = \frac{CR}{\epsilon}. \tag{7}$$

The results of this section are necessary to derive the maximum velocity at which a person or robot can move the detector.

3.1.2.3. Velocity Derivation

The US DOE mandates that SNM handlers perform radiation surveys periodically to prove all removable surface contamination (i. e. plutonium dust) remains at activity levels below $500\frac{dpm}{100cm^2}$ [7]. Surveys must ensure that any contamination at or above this activity are perceived by the RCT and recorded for later remediation. The requirements in 3.1.1 are conservative guidelines to ensure compliance. Requirements such as the distance or the radiation level are limited by the nature of radiation and its ability to interact; however, the velocity requirement is based on hardware and can thus be improved to allow more flexibility.

The source requirement is given in $\frac{dpm}{100cm^2}$ which indicates that the RCT uses a probe of $w\ cm \times l\ cm$ to perform the count and then compares it proportionally to a 10cm by 10cm area. Considering that the RCT will move only along the direction of the width or length of the detector (non-componential), detectors with either $w > 10cm, l > 10cm$ will require fewer cpm to detect the specified contamination levels. This enables the RCT to move more quickly over an area while still detecting the same amount. If the detector is smaller, the RCT will have to move more slowly.

As an example, consider the Ludlum 43-1 alpha scintillator. It has an active area of 83 cm^2, and an efficiency of 33% for Pu-239. Rearranging (7), the count rate at the detector corresponding to 500dpm is

$$CR = \frac{500\frac{dpm}{100cm^2}}{3} = 166.67\frac{cpm}{100cm^2}. \quad (8)$$

The count rate is adjusted for the linear, non-componential travel such that

$$CR = 166.67\frac{cpm}{10cm} \quad (9)$$

assuming square areas. That is, an RCT moving a probe at the required 5cm/s can expect to detect 2.78 counts along a 10cm path. For a detector with an area of $83cm^2$, the same velocity will yield

$$\frac{CR}{\sqrt{83}cm} = 166.67\frac{cpm}{10cm}$$
$$CR = 151.84cpm \tag{10}$$

which would be reported below the specified contamination level if the RCT moved at the same velocity. To account for the change, the RCT simply decreases velocity, moving now at

$$\frac{151.84cpm}{v} = \frac{166.67cpm}{5cm^2}$$
$$v = 4.56\ cm/s \tag{11}$$

which is the maximum velocity the robot can go to ensure a valid survey. A custom sensor with larger dimensions could move faster according to these calculations. It is also important to note that unlike a human RCT, a robot can accurately perceive the measurement rate, and maintain

3.1.3. Technician Survey Methods

Currently, RCTs survey for alpha radiation with a push cart carrying an alpha scintillation detector shown in Figure 3-1.

Figure 3-1: Radiation survey cart in use at LANL [63].

The probe is attached to the bottom of the cart and extends past the wheels to guarantee coverage. The RCT takes readings with the attached counter and records them on a standard form. The survey forms are structured as spreadsheets with entries for time, location, reading, and others.

To survey an entire room, RCTs have to plan a path through the area and step through it recording information as they go. Depending on the size and number of survey areas, this process can take all day[1]. When the area is large or complicated, the technicians can lose track of their progress in the space and must then retrace sections of the path or they may also fail to complete the coverage without realizing it. Another concern is bias towards negative readings. Because the RCAs of interest should never have any contamination, RCTs are conditioned to expect the absence of contamination. The conditioning can cause boredom or lack of attention when

[1] At 5cm/s the RCTbot can traverse a 10m path in 200 seconds. With a 10cm x 10cm detector and accounting for navigation imperfections such as turning and obstacle avoidance, the robot will survey a 10m x 10m room in approximately 6 hours. Given a hallway 30m long by 5m wide with four 10m x 10m rooms, the RCTbot will save an operator approximately 33 hours of surveying time.

surveying, which in turn leads to the neglect of the distance and velocity requirements. A robotic system can address both of these human complications.

3.1.4. Frequency

At the present, RCTs complete surveys under specific circumstances. One reason for a survey is the detection of a problem via other means. (i.e. air monitors detect the presence of contamination in a RCA). A second reason occurs when RCTs change the classification levels of RCAs. An area previously classified as a RA must be fully clear before it can become a RBA. Finally, quarterly scans insure compliance with federal guidelines. The scan provides data to observe trends and serves as a review of the competence of the contamination control program. The requirements are a trade-off between ALARA principles and resource limitations.

As alpha radiation can be undetectable by other means such as air monitors, there remains a possibility that a breach in a RCA could go undetected for some time. The contamination could be spread to many more places before the next routine survey and potentially harm workers. More frequent alpha surveys reduce the probability of contamination spreading.

Scans are also sometimes necessary or desired in areas that have other types of radiation. If an area was large enough, the time taken to survey it could cause RCTs to receive unacceptable doses. In these situations, the survey may be completed over several days, increasing the likelihood of recording errors, or give more time for contamination to spread

The RCTbot can survey around the clock, maintain accurate maps of which areas have been recently surveyed, keep RCTs informed of real-time contamination breaches, and monitor for trends in the data. It also saves time for RCTs to accomplish other necessary tasks related to worker

safety. RCTs must still maintain the instrumentation, respond to contamination alerts, and handle the administrative responsibilities, but the time saved from the survey itself and the increased amount of data for trends should better enable them to protect workers. The next section will provide details about how the RCTbot performs the radiation survey using CCPP algorithms.

3.2. COMPLETE COVERAGE PATH PLANNING

Section 2.2 presented the most current literature on relevant CCPP algorithms. The next section will explain how CCPP is used in the radiation survey task and the decision process for determining which algorithm to apply. The goal of the system is to complete the survey task without adding to the RCTs normal workload while meeting the survey count/velocity and frequency requirements defined above. If the system complicates technicians overall work more than helping, they will stop using it.

For the radiation survey task, it is more important that a robot assures completeness than necessarily performing all measurements in the survey space. RCTs cannot adequately guarantee completeness of knowledge about what has been measured in the space, but they can successfully take measurements in areas where the robot cannot. An ideal robot surveys all parts of a real environment with a fixed detector. An acceptable robot surveys all reachable parts of a real environment and accurately reports which areas it cannot. The goal of an autonomous system for this task is to reduce the time an RCT spends doing survey tasks while upholding current standards.

3.2.1. Online vs. Offline Coverage Planning

The first aspect to consider when determining an algorithm is the type of navigation planning. Section 2.2 introduced the concepts of online and offline path planning and their

meanings with regard to coverage. The main difference between them is how the robot uses its sensors. In online planning, the robot receives information from sensors and reactively makes navigation decisions based on what it believes about the environment. In offline planning, the robot receives a map of the environment to generate a plan and uses its sensors to adhere to this plan. The limitations of the survey task dictate which type of methods a survey robot must implement.

Robots using online planning navigate with equivalent precision in static and dynamic environments because the sensor history is deleted immediately after processing and all information is new. Unfortunately, this also means that the robot is never certain about the true state of the environment unless it is observing it. When trying to survey all of the rooms in a hallway, the robot can only observe open doorways. A great online algorithm can ensure that each of the rooms with an open doorway are completely covered. It cannot provide information to the user about the rooms that were closed off while still reporting that the space was fully covered. Worse, if one doorway closes during the survey, the robot forgets that the room existed at all and still reports full coverage. This is not acceptable for radiation surveys where complete knowledge of the contamination state of the whole environment is strictly mandated. The inability of online planning methods to guarantee complete coverage of the environment restricts the choice of algorithm to offline methods.

Offline methods know the spatial state of the environment as a prerequisite. This means that despite paths being blocked, or changes to certain areas, a robot is always aware that a space that should be coverable was not available. This information can be easily relayed to a technician who can then manually survey the area. It is not ideal in that an RCT has to survey parts of the space, but it is better than losing the knowledge that these parts could still have contamination present.

3.2.2. Algorithm Comparison

Because online methods cannot guarantee coverage, the RCTbot algorithm derives from offline techniques. Section 2.2.2 discusses a number of offline methods such as wavefront propagation, Spiral STC, Morse decomposition and others. Criteria for determining the best include optimality, computation cost, memory cost, and legibility.

Optimality becomes important when considering both resources used by the robot and the frequency of scanning. In larger areas, battery life becomes important, and optimal paths can extend the area covered between recharging. Paths that limit retracing also decrease the time for a single survey and can help improve scan frequency.

Computation costs for algorithms are important for a number of reasons. The batteries of mobile systems limit the power of the computers onboard. In many robots, the computation related to navigation requires most of the processing power. The RCTbot has other computational processes such as data collection, obstacle avoidance, radiation calculations, and spatio-temporal planning, so the cost of path planning must be as low as possible.

Memory costs relate back to the requirement of a static map for offline methods. The amount of memory required for a robot depends on the resolution of the map and the amount of area being covered. A $10m\ x\ 10m$ room decomposed into a grid of mm^2 requires as much memory as a $10km\ x\ 10km$ room in a m^2 grid. The number of rooms in memory also factors into the costs.

Finally, the legibility of the algorithm is important for the technicians who have to analyze the data collected by the robot. The correspondence between the robot's map and the real environment can affect how RCTs perform measurements that the robot cannot handle. If an algorithm uses complicated decomposition techniques or navigates an unpredictable path, the

technician may not be able to accurately determine which areas remain to be surveyed. This is also important if a trend is found in the data indicating a recurrent hazard. The technician must be able to identify easily the source of the trend from the data.

3.2.3. RCTbot Algorithm

Based on the criteria for choosing an algorithm, the RCTbot implements a form of wavefront propagation. While wavefront does not optimize as well as Spiral STC, it performs better than Morse decomposition. It is not computationally expensive at runtime as the path can be generated before motion and stored as a list of cell coordinates for navigation. The memory cost of wavefront is higher than Morse decomposition but similar to Spiral STC because each cell is the same size regardless of the complexity of the environment. Finally, wavefront is the most legible for RCTs because the way the grid is formed and structured resembles how current technicians break down a room and survey it. The current design of the survey supports the style of record-keeping that wavefront propagation naturally produces. Technicians would not need to learn a new system or update any old records to match with the new system. This gives wavefront propagation a huge advantage over other offline methods for this specific task.

The RCTbot must account for the limitations of wavefront propagation. The greatest difficulty is the cost of memory for large areas. In this implementation, each room must be loaded into the robot manually, but in the future, the robot will manage data on the different areas of a facility on a hierarchical basis. Section 3.3 provides the implementation details of the algorithm and describes the wavefront propagation in more detail.

3.2.4. Verification of Completeness

The final challenge of autonomous radiation surveying is verifying that the reported coverage is as accurate as the robot believes. The robot reports coverage based on a number of sources, but each of these sources are internal to the system and subject to bias. Odometry errors are common in navigation, sensors such as SONAR and LIDAR have inherent uncertainties, and modern mobile navigation techniques are all based in probabilistic methods. Technicians must calibrate the robot's navigation in a similar manner to calibrating radiation instruments. The measured data must be compared to known, external data. In order to obtain this data, methods of externally monitoring the robot during a coverage task must be implemented.

Metrics for coverage performance have previously been developed by Wong *et al* [65]. Using external cameras and vision processing, the coverage achieved by a robot can be confirmed by external sources. These metrics help evaluate the RCTbot's navigation and coverage system.

3.3. RCTBOT COVERAGE ALGORITHM

Section 3.2 discussed the theory of CCPP, and 3.2.3 in particular discussed the wavefront algorithm. The robot performs a three-part process to execute a wavefront CCPP survey from a static map. The three parts are discretization, wavefront propagation, and path execution. The next sections explain the implementation of each part in detail.

3.3.1. Discretization

The robot receives a map in the form of a black and white picture where black indicates non-traversable space. For a grid-based planning method such as the wavefront algorithm, the

resolution of discretization affects the overall coverage plan. Resolution dictates the size of individual grid cells and how they fill the space. Either a macro-cell or micro-cell option is possible. A macro-cell generally outlines the footprint of the robot; however, it can be any size as long as aligning the center of interest (i. e. the center of the sensor) with the center of the macro-cell results in coverage of the cell. Conversely, micro-cells are smaller than the footprint of the robot. The robot can cover more of the micro-cells in complex environments. Figure 3-2 shows an example of an environment with a macro-cell and micro-cell discretization.

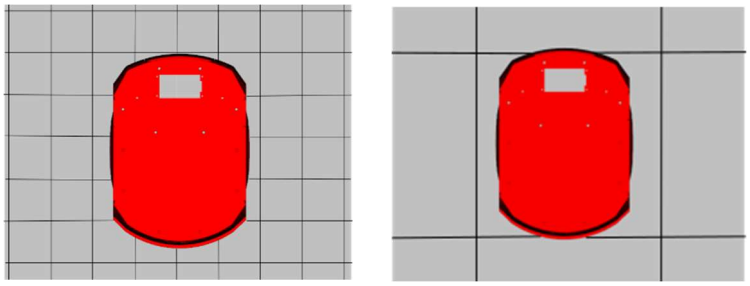

Figure 3-2: Example of micro-cell vs. macro-cell decomposition.

The robot cannot navigate its center in certain hard to reach areas, leaving macro-cells near walls and in corners difficult to cover. The sensor reaches more of the same area if it is discretized into micro-cells, leaving less for a human RCT to cover afterward.

The majority of grid-based plans assume macro-cells because it is simpler to calculate the optimal navigation path. Because any point that is on the robot's sensor (not just the center point) could cover a micro-cell in the map, it is a difficult challenge to plan micro-cell coverage. It is also more memory intensive to plan for all micro-cells than macro-cells. For a radiation survey, a compromise is to use the detector's footprint for navigation instead. Placing the center of the detector over a point of possible contamination is necessary to obtain the most accurate reading. For this work, coverage planning will use macro-cells based on the footprint of the detector. In

future work, the plan will use a hybrid cell approach such as in [66] where areas closer to complex obstacles will have a denser population of fine resolution cells, and areas in free space will use the detector footprint.

3.3.2. Wavefront Propagation

Once the robot has a fully discretized map, it applies the wavefront algorithm to generate a coverage navigation plan. Section 2.2.2 outlines the concept of wavefront propagation. The robot designates a start point and a goal point (usually a cell near the start or in some other strategic position like a charging station). It labels the goal as the zeroth layer. Then it increments the layer number in each adjacent, unlabeled cell until the space is completely filled. The wavefront algorithm assumes macro-cell discretization when ordering the cells in the path plan.

3.3.3. Plan Execution

It executes the path by navigating from one cell in the path plan to the next until it has surveyed all of the reachable cells. Figure 3-3 shows a decision tree for the process. After the robot begins surveying, it can only terminate by returning to the dock after completion or awaiting operator instructions. Naturally, error states could occur where the robot is unable to complete the plan, but these events are not included in the plan for execution. Two of the decisions in the tree concern whether to continue scanning or terminate and return to the dock and the others involve informing the operator.

The first layer of decisions check to see if the robot can possibly cover more cells after its initial attempt. This accounts for dynamic obstacles that might have occupied a cell during the first scan such as a person or a cart. The likelihood of coverage mentioned in the block on the left side

relates to the occupancy prediction for cells the robot could not cover. Cells with a prediction near, but below the threshold for the static map are much more likely obstructed than cells sufficiently far below probability. The robot would return to investigate cells that had a low prediction probability and ignore those with a higher value. The tendency of the robot to return to difficult areas additionally provides more data where it might have lacked previously.

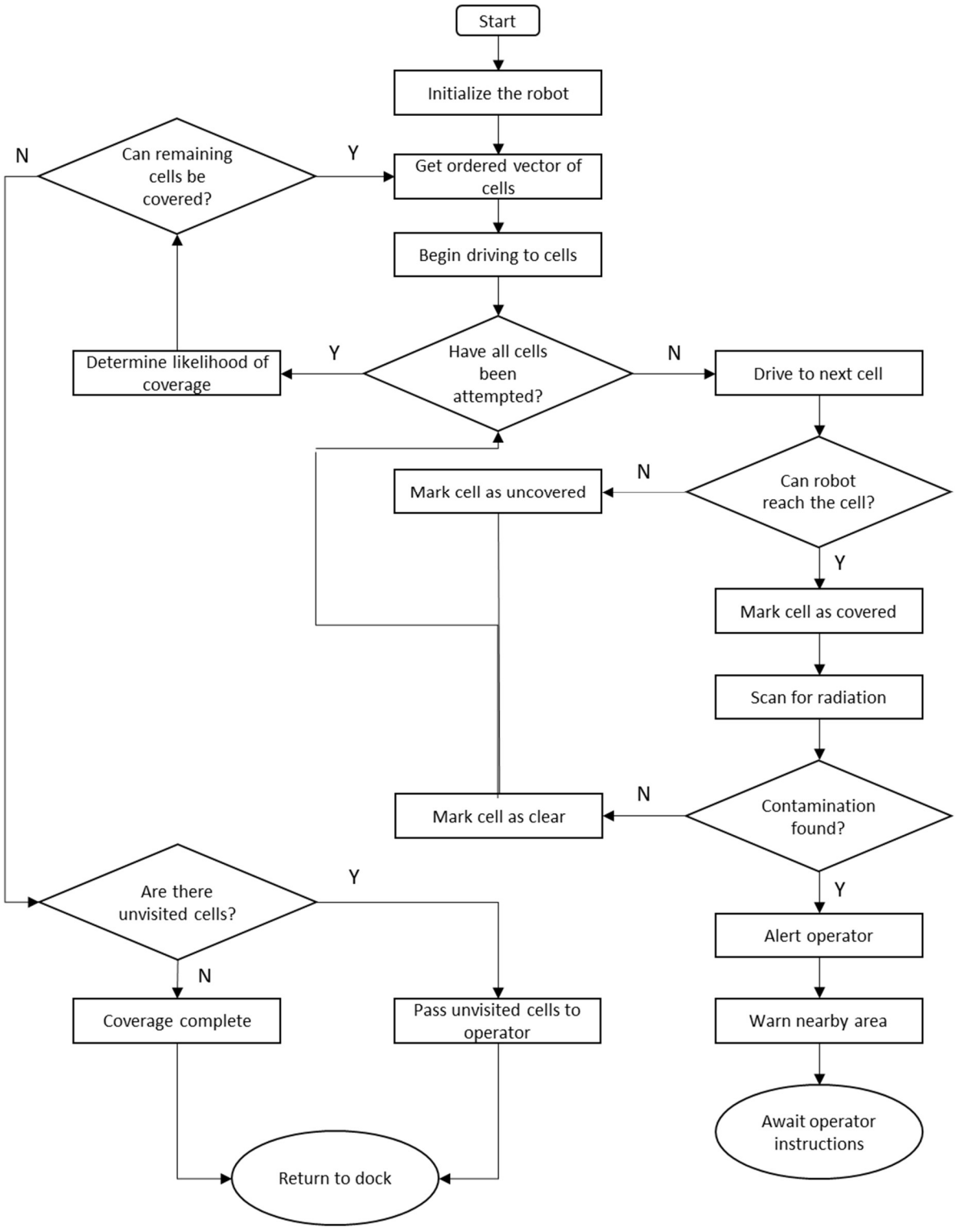

Figure 3-3: Decision tree for the CCPP algorithm.

The second set of decision questions relate to the robot-operator interactions. The first occurs during the survey stage where the robot discovers contaminations and must alert its operator to the problem. The other decision only takes place after the robot attempts to cover all possible cells, accounting for the likelihood of coverage. The robot either reports the unvisited cells to the operator or returns to its dock until the next scheduled survey. It is important that the robot informs the operator of the uncovered space in case it persists into future surveys and an area is continually uncovered.

3.4. SUMMARY

This chapter demonstrated the necessity and utility of a robot for radiation surveys. The RCTbot implements wavefront propagation to execute complete radiation area coverage. It is able to perform radiation surveys with more frequency, precision, and speed than a human. It works alongside RCTs, handling the majority of surveys so that RCTs can focus on other important tasks without sacrificing worker safety.

Chapter 4: Spatio-temporal System Explanation

The previous chapter describes the task specific challenges of radiation surveying with the RCTbot. This chapter describes the theory of STMM and how a robot uses it to update its environment map and improve its long-term autonomy. The overall goal of STMM is to keep the robot's map comparison close enough to the true state that it can still be used in navigation.

4.1. Realistic Environments for Robot Operation

As previously mentioned in Chapter 1, real environments have spatio-temporal characteristics. That is, the physical layout of the environment changes over time and nothing remains indefinitely static. The rate at which the environment changes is important for decision processes in autonomous systems.

4.1.1. High and Low Dynamics

Researchers describe the spatio-temporal state of a system in terms of dynamics. A low dynamic environment is one that changes with low frequency compared to a specified frame of reference, and a high dynamic environment is one that changes more frequently. For autonomous radiation surveys, the frame of reference is the time to complete scans. If a scan area has many changes during a scan, then it is a high dynamic area. If the area changes very little during a scan, then it is low dynamic. A semi-static environment changes minimally between scans.

RCTs perform radiation contamination surveys in generally low dynamic areas. RCAs are tightly regulated, and most large obstacles that could affect localization scores are permanent fixtures that will seldom change position. Due to the low dynamics, RCTbot can approximate the

environment as static. So long as the changes between surveys still allow for localization, the robot can relay any failures to survey an area back to the RCT and the survey can still be useful. Otherwise, the system could fail to localize and any survey data obtained would be illegible to technicians.

4.1.2. Adaptive Permanence

The environment as a whole has a certain type of dynamics, but objects within it can differ in how permanent they are. Some objects such as walls, cabinets, columns, and doorframes are generally static. Other objects, such as forklifts, gloveboxes, and doors are not necessarily static in the environment, but exhibit varied degrees of permanence. Furthermore, the permanence of any obstacle in the environment can adapt over time. When a new cabinet is installed, the robot will believe it to be temporary for a while. Over time, its belief in the cabinet's permanence changes to reflect the true state. Likewise, if a wall that has been around for many years is torn down in a construction project, the robot might believe its sensors have simply failed to detect it. Over time, the permanent state of the wall diminishes though because the robot receives more evidence that the object no longer exists.

Static elements assist the robot in maintaining an accurate spatio-temporal map of the area. These objects are often prominent features used for localization, and their tendency to remain static bolsters the likelihood the robot will successfully localize despite small changes in other obstacles.

Semi-static objects are also useful. Forklifts might follow a schedule that intersects with a survey at certain points. Doorways could be open or closed depending on the time of the day. While the spatio-temporal state of these obstacles is not static, it is still informative. Patterns and

trends in these semi-static objects can inform navigation decisions. For example, if a door to a room is opened and closed many times on a certain day of the week, but left open all the time on another day, the robot can plan to visit that room when it is most accessible. If a forklift works in a certain part of a factory floor on the first week of every month, avoid that area during the time.

Adaptive permanence in objects in the map can be leveraged to make more intelligent navigation decisions. This helps improve the optimality and legibility of the algorithm and increases the robot's usefulness for technicians.

4.2. PROBABILISTIC UPDATING

The previous section detailed the importance of spatio-temporal updating for maintaining the RCTbot's ability to complete its task. This section gives a brief overview about how the robot performs the update process through data collection and map updating.

4.2.1. Data Collection

The data collection method for spatio-temporal updating is completely dependent on the robot's navigation sensors. The robot's sensors collect vast quantities of data when performing localization. This data is compared to the static map and then removed from memory. Spatio-temporal updating requires saving the collected data, filtering it with probabilistic methods, and generating a snapshot of the map after each scan. As a motivating example, consider the area shown in Figure 4-1.

Figure 4-1: Storage environment as expected (left) and with an unexpected obstacle (right).

The storage area is usually free of any semi-static obstacles. Between scans, someone has placed a trash can. As the robot surveys the area, it develops the data shown in Figure 4-2.

Figure 4-2: Storage area with robot's static map (black), perceived map (yellow), and probabilistic map (orange).

As it moves through the environment, it localizes by comparing the yellow of the data points to the black data that is the static map. After visiting an area, the points that have been filtered and saved in memory remain as the orange. Figure 4-3 shows the robot's static map, or belief about the environment side by side with the latest completed scan.

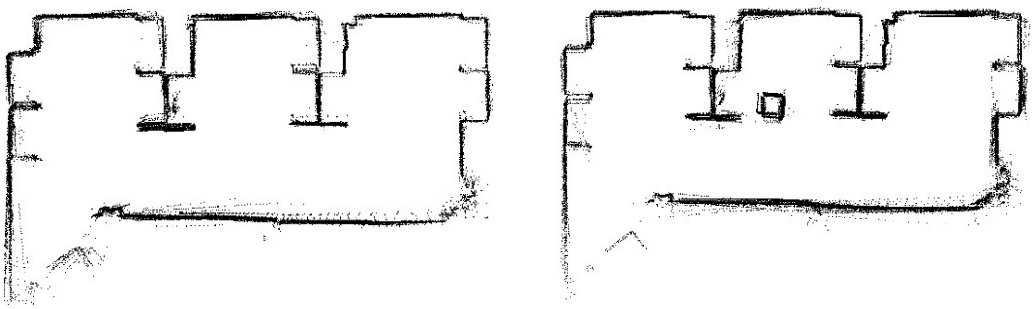

Figure 4-3: Perceived map data from storage area with and without obstacle.

The majority of the objects remained the same. The robot still managed to localize despite the presence of the new object in the middle shelves, and so it completed the scan while reporting the area beneath the trash can as uncoverable. In many cases, the trash can will be removed before the next scan, and the static map will remain accurate enough, but other objects like a newly installed cabinet might not. Any small changes in the perceived obstacles reflected in data must be stored and analyzed by the system in order for the robot to continue reliable operation.

4.2.2. Probability Map

This effort has established that generated maps do not usually match the reality, but the disparity can be addressed using *probability maps*. A probability map is used to contain the robot's beliefs about the state of the environment. It contains information about which obstacles most likely exist at the beginning of a scan and also how likely they are to exist in the data collected after the scan. Based on typical robotic sensor data, the probability map is list of points in the environment and their corresponding probabilities of occupancy. Consider the example in Figure 4-4.

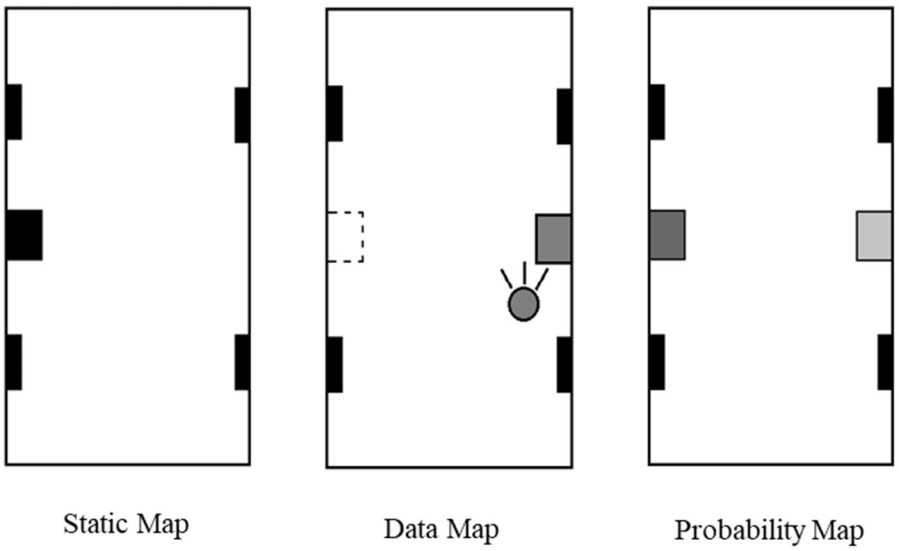

Figure 4-4: Perceived map data from storage area with and without obstacle.

The static map on the left shows a room with three shelves and a larger cabinet. When the robot next scans the room, the cabinet has been moved to the opposite wall. The shelves still allow the robot to localize, but the cabinet has changed its spatial state. From this observation, the robot generates the map on the right. It creates a set of points representing the shelf on the right side of the room and assigns a low confidence value to it. The robot also decreases its belief about the set of points representing the obstacle on the left side of the room. Both sets exist in the robot's probability map, but one is more likely than others. Semi-static objects exist in many possible locations, so they reoccur in the probability map. More permanent objects are reconfirmed with every scan, reinforcing the robot's belief in their existence. It takes much longer for a wall to be removed from the robot's memory than a cart.

The probability map is consulted before each scan to determine what the most likely static representation of the environment will be. This maximizes the chance that the robot will

successfully localize and complete the survey. The next section explains how the robot's data is used to develop the probability map between scans.

4.3. BAYESIAN SURVIVAL ANALYSIS

To make intelligent decisions about how to populate the static map, the robot must estimate how long each collection of points in an object will *survive,* or remain spatio-temporally static. The robot decides based on the probability that each point comprising the object remains occupied space in subsequent surveys. This work generates estimations based on previous literature where Bayesian Survival Analysis (BSA) characterizes feature persistence [59]. BSA, described in [67], involves developing probabilistic prior distributions based on the relative *survival time* of features in the environment. Survival time describes the duration of existence of a feature in a real environment. It helps the robot predict the true state of occupancy for each point between surveys by predicting how long each point remains an obstacle in the probability map. The following section details the theory of BSA based on Rosen and used for STMM.

4.3.1. Environment Modeling

Consider the laser range data surveyed by the robot as individual points. Each data point encodes information about its occupancy state as a Boolean random variable – the point is occupied or not. Throughout this analysis, a single point in the map is considered but the algorithm can be extended to evaluate a vector of points that comprise the probability map.

The state for a single point can be modeled as

$$X_t \in \{0,1\}$$

$$\{t_i\}_{i=1}^{N} \subset [0,\infty) \tag{12}$$

where X_t is a variable encoding the true occupancy state of the point as a Boolean value at each time t. The robot takes measurements over N intervals to obtain

$$\{Y_{t_i}\}_{i=1}^{N} \subseteq \{0,1\} \tag{13}$$

where Y_t represents the noisy sensor observations.

The relationship between the true occupancy state, X_t, and the observations of that state, Y_t, is given by

$$Y_t|X_t \sim p_{Y_t}(\cdot\,|X_t) \tag{14}$$

where $p_{Y_t}(Y_t|X_t)$ is a conditional probability distribution modelling the uncertainty in the detection system. Two types of uncertainty govern the distribution for sensors: the probability of missing a point object given that it is really present, P_M, and the probability of reporting an absent point object as a false positive, P_F, formalized as

$$P_M = p_{Y_t}(Y_t = 0\,|X_t = 1)$$

$$P_F = p_{Y_t}(Y_t = 1|X_t = 0)$$

$$P_M, P_F \in [0,1] \tag{15}$$

Both assume that the uncertainty represented is systematic to the detector, and therefore constant throughout the analysis. The next section introduces the framework for estimating survival time in the system described above.

4.3.2. Survival Time in a Bayesian Framework

Let $T \in [0, \infty)$ be the survival time of a point object. The conditional relation to the occupancy state is

$$X_t|T = \begin{cases} 1, & t \leq T \\ 0, & t > T \end{cases} \tag{16}$$

where t is described in (12). For all times up to and including T, the point object survives, and for all times after T, the object ceases to appear in the environment.

The robot's goal then, is to predict X_t for all points in the next scan given the past information on whether a point has survived until that moment and evidence from the robot's sensors. Predicting X_t based on prior knowledge and sensor evidence suggests the use of Bayesian inference. Bayes' Rule is given by

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{17}$$

where $p(A|B)$ is called the posterior probability, $p(B|A)$ is called the likelihood, $p(A)$ is the prior probability, and $p(B)$ derives from data or evidence, which is often taken as a normalization factor because it does not depend on the state variable A [14] [68]. In terms of the model from Section 4.3.1, the random variable for observations, Y_t, is realized as

$$y_{1:N} \triangleq \{y_{t_i}\}_{i=1}^{N} \tag{18}$$

and the random variable for the true state, X_t, is realized as

$$X_t = 1 \tag{19}$$

for all times of interest through time $t = T$. To interpret Bayes' Rule in terms of the model, it becomes

$$p(X_t = 1|y_{1:N}) = \frac{p(y_{1:N}|T \geq t)p(T \geq t)}{p(y_{1:N})} \tag{20}$$

where it is implied from (16) that X_t shares a relationship with T such that the likelihood and the prior are written in terms of survival time.

The prior distribution in (20), $p(T \geq t)$, is best represented by a Probability Density Function (PDF). Consider the following PDF, p_T, that defines the prior distribution of T.

$$T \sim p_T(\cdot) \tag{21}$$

Section 4.3.4 specifies the exact form of the distribution, but the function

$$F_T(t) \triangleq p(T \leq t) = \int_0^t p_T(x)dx \tag{22}$$

generally describes the distribution and allows the prior in (20) to become

$$p(T \geq t) = 1 - F_T(t) \tag{23}$$

by letting the integral in (22) go to 1 as $t \to \infty$. Additionally, the likelihood function generalizes to

$$p(y_{1:N}|T) = \prod_{t_i \leq T} {P_M}^{1-y_{t_i}}(1 - P_M)^{y_{t_i}} \prod_{t_i > T} {P_F}^{y_{t_i}}(1 - P_F)^{1-y_{t_i}} \tag{24}$$

in terms of the error parameters P_M and P_F. This indicates that the probability of the sensor measurements being correct given the known survival time, T, of the point object depends on the product of the probabilities of missing the object in each scan before T and the probabilities of falsely sensing the presence of the object in each scan after T.

Consider $t \in [t_N, \infty)$, or any time after the last measurement. Using (24) with the likelihood for the case that $T \geq t$, yields

$$p(y_{1:N}|T \geq t) = p(y_{1:N}|t_N) = \prod_{i=1}^{N} {P_M}^{1-y_{t_i}}(1 - P_M)^{y_{t_i}} \tag{25}$$

which summarizes all possible errors before $t = t_N$. Finally, notice that the right-continuous function from (24) is constant on the intervals $[t_i, t_{i+1}) \; \forall \; i = 0, \ldots, N$ if $t_0 \triangleq 0$ and $t_{N+1} \triangleq \infty$. By integrating across all values of T, the evidence term in closed form develops as

$$\begin{aligned} p(y_{1:N}) &= \int_0^{\infty} p(y_{1:N}|T) \cdot p(T) dT \\ &= \sum_{i=0}^{N} \int_{t_i}^{t_i+1} p(y_{1:N}|t_i) \cdot p(T) dT \\ &= \textstyle\sum_{i=0}^{N} p(y_{1:N}|t_i) \cdot [F_T(t_{i+1}) - F_T(t_i)] \end{aligned} \tag{26}$$

where (22) defines F_T. Substituting (23), (25), and (26) into (20) yields

$$p(X_t = 1|y_{1:N}) = \frac{p(y_{1:N}|t_N)}{p(y_{1:N})}\big(1 - F_T(t)\big), \qquad t \in [t_N, \infty) \tag{27}$$

which is the Bayesian form of the survival probability for a single point object.

4.3.3. Recursive Bayesian Estimation

Now that a term for the posterior probability exists through BSA, further analysis partitions the Bayesian expression into an algorithm for recursive estimation. The recursive algorithm must calculate the posterior $p(X_t = 1|y_{1:N})$ over $t \in [t_N, \infty)$ to demonstrate that the step past the current time t_N is taken effectively as all time after t_N. In this manner, the robot recovers a

posterior estimation after each scan to be factored into the probability map in time for the next plan's development.

The first step is looking at the transition from $t = t_N \rightarrow t_{N+1}$ and the corresponding effect on $y_{1:N} \rightarrow y_{1:N+1}$ in the likelihood and evidence terms. The difference between the parameters $y_{1:N}$ and $y_{1:N+1}$ in terms of (24) is

$$p(y_{1:N+1}|T) = p(y_{1:N}|T) \times \begin{cases} {P_F}^{y_{t_{N+1}}}(1-P_F)^{1-y_{t_{N+1}}}, & T < t_{N+1} \\ {P_M}^{1-y_{t_{N+1}}}(1-P_M)^{y_{t_{N+1}}}, & T \geq t_{N+1} \end{cases} \tag{28}$$

such that, reinforced by (25), the step iteration of the likelihood term becomes

$$p(y_{1:N+1}|t_{N+1}) = {P_M}^{1-y_{t_{N+1}}}(1-P_M)^{y_{t_{N+1}}} p(y_{1:N}|t_N). \tag{29}$$

This suggests the use of a loop-based algorithm where the step iterator is the time as it transitions from $t = t_N \rightarrow t_{N+1}$.

Likewise, the evidence term determined in (26) iterates using the lower partial sum

$$L(y_{1:N}) \triangleq \sum_{i=0}^{N-1} p(y_{1:N}|t_i)[F_T(t_{i+1}) - F_T(t_i)] \tag{30}$$

and extracting (30) from (26) to evaluate only at N as

$$\begin{aligned} p(y_{1:N}) &= p(y_{1:N}|t_N)[F_T(t_{N+1}) - F_T(t_N)] \\ &= p(y_{1:N}|t_N)[1 - F_T(t_N)] \end{aligned} \tag{31}$$

because any time after t_N is taken to be

$$t_{N+1} \triangleq \infty \tag{32}$$

and from (22)

$$F_T(t_{N+1}) = F_T(\infty) = \int_0^\infty p_T(x)dx = 1 \tag{33}$$

on a continuous, cumulative distribution. Substituting (30)-(33) into (26) yields

$$p(y_{1:N}) = L(y_{1:N}) + p(y_{1:N}|t_N)\,[1 - F_T(t_N)]. \tag{34}$$

When transitioning from $L(y_{1:N}) \rightarrow L(y_{1:N+1})$ the final summation term from (30) combines with (28) given $T < t_{N+1}$ and the value of $p(y_{1:N})$ to yield

$$L(y_{1:N+1}) = {P_F}^{y_{t_{N+1}}}(1 - P_F)^{1-y_{t_{N+1}}} \times$$
$$(L(y_{1:N}) + p(y_{1:N}|t_N)\,[F_T(t_{N+1}) - F_T(t_N)]) \tag{35}$$

where the false alarm probability term, P_F presented in (15), accounts for the chance of survival through the time iteration. The iterated lower partial sum from (35) substitutes into (34) to give the iterated evidence term. Using the iterated evidence term, the iterated likelihood from (29), and the cumulative distribution of the prior, the Bayesian form in (27) calculates the posterior probability for the step iteration.

The next section presents the theoretical method of determining the prior distribution of the state that was left in general form in (22). Chapter 5 details the precise implementation of the algorithm on the robot for temporal map updating.

4.3.4. Generation of Prior Distribution Functions

The BSA algorithm affects the posterior estimation to the greatest degree in the generation of prior distribution function. In (22), the cumulative distribution for p_T is defined by the integration from 0 to *t* of $p_T(x)$, wherein the calculation determines the total probability that the object does not survive until the present time, or $T \leq t$.

The *survival function* represents a new distribution where $T > t$ represented by

$$S_T(t) \triangleq p(T > t) = \int_t^{\infty} p_T(x)dx = 1 - F_T(t) \tag{36}$$

where it examines the probability of continued survival after the present time *t*. Because survival time requires predicting the probability of survival past known data, the robot must have a way to estimate how hazardous a step of time is to various objects in the environment. The *hazard*, $h_T(t)$ is a measure of the probability of an object failing to survive each additional time step. It is given by

$$h_T(t) \triangleq \lim_{\Delta t \to 0} \frac{p(T<t+\Delta t|T \geq t)}{\Delta t}. \tag{37}$$

The conditional probability in (22) can be rewritten as the union of the two parameters over the condition such that it becomes

$$h_T(t) \triangleq \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{p(t \leq T<t+\Delta t)}{p(T \geq t)}. \tag{38}$$

By converting the hazard into an implicit *hazard rate* or *hazard function*, and by implementing the definitions in (22) and (36), the term becomes

$$\lambda_T(t) = \frac{p_T(t)}{S_T(t)}. \tag{39}$$

which is equivalent to

$$\lambda_T(t) = -\frac{S_T'(t)}{S_T(t)}. \tag{40}$$

From this, we can state that

$$S_T(t) = e^{-\Lambda_T(t)} \tag{41}$$

where

$$\Lambda_T(t) \triangleq \int_0^t \lambda_T(x)dx \tag{42}$$

is the *cumulative hazard function*. Upon differentiating both sides of (41), the prior distribution term becomes

$$p_T(t) = \lambda_T(t)e^{-\Lambda_T(t)}. \tag{43}$$

By writing the prior in terms of only the hazard function, the challenge of BSA is now to estimate how time affects the survival different objects in the environment.

The prior is the only tunable parameter in the BSA algorithm. The evidence comes directly from the robot's sensors, and the likelihood is a function of the sensor readings. Constructing the prior with the hazard function strongly influences inferences made by the BSA algorithm. Adjusting the hazard function designs the prior such that it more accurately describes the true state of the environment and improves predictions.

Rosen proposes three methods for designing the prior via the hazard function [59]. The first method proposes a semi-non-informative prior where the survival function undergoes exponential decay. The second method, which designs an informative prior, examines how object recognition and classification could be used to apply class-specific hazard rates and predict the characteristic survival times of different objects. The third method uses a pure non-informative prior constructed via further Bayesian analysis on the hazard function to estimate the distribution over $\lambda_T(t)$ for various objects. This work employs the third method because it has the property of *minimal informativeness*, and thus is most likely to reduce introduced bias. In future efforts, the other methods may prove more useful provided that true estimates of the hazard rate can be obtained.

In the third method, the algorithm's goal is to find a general-purpose prior that is minimally informative, such that it can be used with as little information about the environment as possible.

The prior assumes uncertainty in the hazard function, and thus explicitly models this uncertainty with another Bayesian prior

$$\lambda \sim p_\lambda(\cdot). \tag{44}$$

This second prior is introduced into the analysis by marginalizing the prior distribution of T by λ

$$p_T(t) = \int_0^\infty p_T(t;\lambda) \cdot p_\lambda(\lambda)\, d\lambda. \tag{45}$$

From (43), the hazard function relates to the prior distribution of T as a rate parameter. Given that survival estimates can be assumed to have bounds on the order of seconds for a lower bound and decades as an upper bound, [Rosen] shows how the algorithm can safely generate the marginalized prior

$$p_T(t;\lambda_l;\lambda_u) = \frac{e^{-\lambda_l t} - e^{-\lambda_u t}}{t\ ln\left(\frac{\lambda_u}{\lambda_l}\right)} \tag{46}$$

and the survival function

$$S_T(t;\lambda_l;\lambda_u) = \frac{E_1(\lambda_l t) - E_1(\lambda_u t)}{\ln\left(\frac{\lambda_u}{\lambda_l}\right)} \tag{47}$$

and $E_1(x)$ represents the first generalized exponential integral

$$E_1(x) = \int_1^\infty t^{-1} e^{-xt}\, dt. \tag{48}$$

This is a general-purpose prior for a rate marginalized prior with minimal informativeness and maximum uncertainty.

Substituting S_T into (27) allows the BSA algorithm to close the recursive loop and calculate the posterior distribution as

$$p(X_t = 1|y_{1:N}) = \frac{p(y_{1:N}|t_N)}{p(y_{1:N})}\big(1 - F_T(t)\big), \qquad t \in [t_N, \infty)$$

$$p(X_t = 1|y_{1:N}) = \frac{p(y_{1:N}|t_N)}{p(y_{1:N})} S_T, \qquad t \in [t_N, \infty) \tag{49}$$

$$p(X_t = 1|y_{1:N}) = \frac{p(y_{1:N}|t_N)}{p(y_{1:N})} \cdot \frac{E_1(\lambda_l t) - E_1(\lambda_u t)}{\ln\left(\frac{\lambda_u}{\lambda_l}\right)}, \qquad t \in [t_N, \infty)$$

which is the final prediction of what the state of the point will be in the next time step. Because the prior is general and non-informative, the only varying factor is the time such that the hazard increases the closer the object is to the upper bound. This means that the prior is uniform in any given time step for all points in the map.

In order to extend this to all data points from the survey, the BSA algorithm is simply run on each point in the map. The probability map contains the output of each point after it is processed with BSA and acts as a distribution for the static map. Before each survey, the robot samples from the probability map for the next cycle and uses the sampled points as the static map.

4.4. SUMMARY

Using only the sensors required to localize, the RCTbot can leverage the temporal characteristics of the environment to predict how it will appear the next time it navigates the space. The ability to track changes in the map and maintain its usefulness over time increases the long-term autonomy of the robot. Predicting the next state of the environment also enables the robot to make intelligent navigation decisions, optimize the radiation coverage, and minimize human RCT

exposure. The next chapter provides details of the implementation including details of the algorithms provided in this chapter.

Chapter 5: Implementation

While previous chapters discussed the theoretical aspects of robotic radiation surveys with temporal map updating, this chapter details the implementation. It includes the hardware, software, and logic of the system and describes the demonstration goals, metrics, and results. Note that this chapter is about a specific implementation of the RCTbot on a given hardware system. Where aspects are platform independent, the section will indicate it for the reader.

5.1. HARDWARE CONFIGURATION

The hardware configuration for the radiation survey comprises all of the sensors, actuators, and mechanical design features such as the robot, its navigation sensors, and radiation sensors. The sensor configuration is applicable to any mobile robot so long as the power and space requirements do not exceed the load.

5.1.1. Adept PioneerLX

The specific platform used for the RCTbot in this work is the Pioneer-LX mobile research platform [69] by Omron Adept Mobile Robots. It is an industrial platform designed for long-term deployment in environments such as factories or warehouses. The system contains built in navigation sensors such as SONAR and LIDAR as well as wheel encoders for odometry and a manual bumper switch for safety. It possesses a 24VDC LiFePO4 battery with 60Ah capacity for a 13h battery life (unloaded) which make it ideal for long-term autonomous tasks. The Pioneer-LX has a payload capacity of 60kg on a flat surface, so it can carry a wide array of secondary sensors. Figure 5-1 shows the NRG's Pioneer-LX system with all sensors equipped.

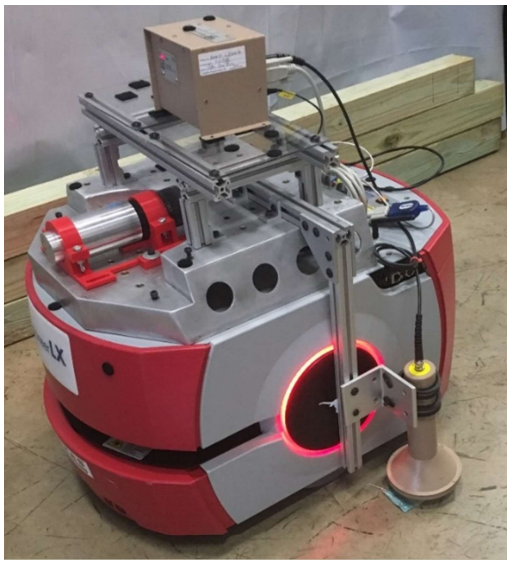

Figure 5-1: Pioneer equipped with radiation coverage sensors.

The robot navigates with two drive wheels located along the horizontal axis through the center of mass and balances on four caster wheels. This differential drive configuration allows the platform to turn with a zero radius, which helps the maneuverability in tight corners. This quality is important for following wavefront generated paths which assume the robot moves from one grid cell directly to an adjacent grid cell. It also provides for a method of ensuring the robot never travels over space that has yet to be surveyed except for its current cell. This helps prevent the spread of contamination if it exists.

5.1.2. Alpha Detection Hardware

As mentioned in Chapter 3, requirements on the radiation survey detector determine the instrumentation hardware. For this work, the hardware configuration included a Ludlum 43-1 probe connected to a Ludlum Single Channel Analyzer (SCA) with 4 channels. Figure 5-1 also

displays the probe and SCA box attached to the Pioneer-LX. In this experiment the robot carried only the one probe to its side. This would be insufficient in a real survey task because of the high likelihood of driving through any contaminated surface before detecting radiation. Instead, the robot should have a detector configuration such as the one shown in Figure 3-1. This configuration sees the detector covering the entire width of the robot and going before the wheels in order to prevent the spread of contamination.

5.2. SOFTWARE CONFIGURATION

Robotic systems inherently combine software with hardware for full functionality. The software required for radiation survey includes robot specific drivers, supervisory logic and planning algorithms, sensor drivers, and user interfaces. This section goes into the details about the software architecture including the underpinning framework, the robot drivers, the planning algorithms, and the radiation detection scheme.

5.2.1. ROS: The Robot Operating System

At the center of every software component in this project is a pseudo-operating system called the Robot Operating System (ROS). ROS builds upon a Unix platform (commonly Ubuntu) and presents a framework for communication, process handling, hardware abstraction, and package management [70]. Beyond its technical aspects, ROS is also a key component of a thriving community of open source robotics software developers. The ROS community maintains tools and libraries for robotic systems across programming languages, hardware configurations, and Unix distributions. This work benefits from the use of the ROS navigation stack, visualization tools,

robot and tool specific drivers written by the community at large, and other generic tools provided by ROS.

5.2.2. Driver Implementations

Despite most of the ROS software being hardware agnostic, one must eventually implement it on a specific system. ROS handles this by using hardware drivers for communication with the individual robots and tools. Often, the drivers act like a wrapper on the control software packaged with the robot in order to translate commands into the ROS communication framework and vice versa. This section discusses how this is handled for the specific case of the Pioneer-LX doing radiation surveys.

The first consideration is the robot drivers. The Pioneer-LX is equipped with two software control packages: ARIA and ARNL. ARIA is a library with packages that communicate with low level functions such as motor commands, encoder readings, sensor data, and the like for easy user interaction and high-level control of the robot. ARNL is an autonomous navigation software suite that receives data from the SONAR and LIDAR sensors, performs SLAM to build maps of areas, and then uses a form of AMCL to navigate in the map. Both of these packages provide user interfaces so that the operator can use the robots directly. In order to link ARIA and ARNL to other useful ROS packages, the community designed drivers, `rosaria` and `rosarnl`, to correspond with each respectively.

5.2.3. BSA Algorithm Implementation

The BSA algorithm stands apart from the coverage process because it is performed entirely offline. After coverage is complete, the sensor data collected during the survey is saved and filtered

to get unique observations of each cell. The BSA algorithm initializes with this sensor data and information about its parameters from the previous calculation. The algorithm is outlined below and draws from the theory presented in Chapter 4.3.

Algorithm 1 BSA Recursion Execution

Input: Prior distribution $p(T > t)$, sensor data y_t.
Output: Posterior probability $p(X_t = 1|y_{1:N})$ for $t \in [t_N, \infty)$.

1: **Initialization:** Set parameters $t_0 \leftarrow 0, N \leftarrow 0, p(y_{1:0}|t_0) \leftarrow 1, L(y_{1:0}) \leftarrow 0, p(y_{1:0}) \leftarrow 0, P_M \leftarrow 0.1, P_F \leftarrow 0.01$
2: **while** ∃ new data $y_{t_{N+1}}$ **do**
Update:
3: Compute lower partial evidence $L(y_{1:N+1})$ using
4: Compute likelihood $p(y_{1:N+1}|t_{N+1})$ using
5: Compute evidence $p(y_{1:N+1})$ using
6: $N \leftarrow (N + 1)$.
Predict:
7: Compute posterior survival probability $p(X_t = 1|y_{1:N})$
8: Update parameters
9: **end while**

The output of the BSA algorithm is the posterior probability of survival, $p(X_t = 1|y_{1:N})$ for each point in the map. The RCTbot iterates through all points and then generates a sample for the static map. Temporally static objects such as wall will contain points with probabilities approaching 1 while temporally dynamic objects such as crates should have probabilities much lower than 1 but not 0. Similarly, perpetually free space will contain those points with probabilities closest to 0. In this manner elements of the environment are more likely to survive the longer they already have.

5.3. EXPERIMENT

The thesis thus far claims the ability of STMM to improve long-term autonomy in radiation coverage survey applications. To demonstrate this ability, this section will outline experiments performed with the robot to accomplish CCPP and STMM and then discuss the implications of those experiments.

5.3.1. CCPP Implementation

Section 3.2 and Section 3.3 discuss details of the CCPP algorithm design and implementation and this section will describe experiments done to establish its efficacy. A CCPP algorithm must demonstrate the ability to perform the basic role of an RCT and provide additional data in order fulfill the requirements enumerated in Chapter 3. In [71], a prototype of the RCTbot shows this by surveying a small area for alpha radiation, record coverage progress, and provide feedback to a human operator. The experimental environment in [71] is shown in Figure 5-2.

Figure 5-2: Test environment space for CCPP experiment [71].

The prototype of the RCTbot used one alpha sensor to survey a very simple environment with two possible obstacles, a box and a cylinder. The space was $12' \times 9'$, and the right side was open during the experiment. The experiment consisted of a series of trials where the robot generates coverage path plans, surveys for contamination, and reports the results of the survey to a remote operator. Obstacles in the test environment appear in the four configurations shown in a-d of Figure 5-3. Figure 5-3 also shows the four coverage paths and resulting execution for each configuration.

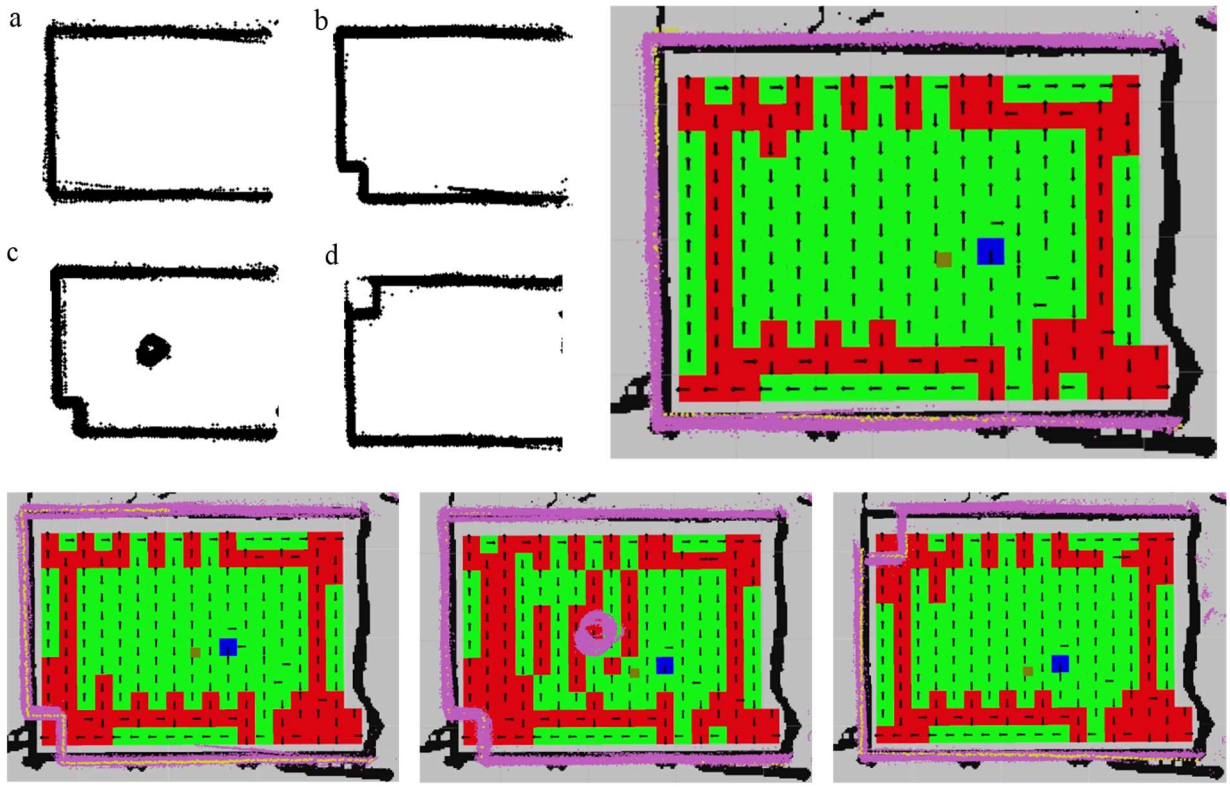

Figure 5-3: Sensor data and coverage path for four contamination survey trials [71].

The green represents covered cells, the red indicates cells which the robot could not reach, and the blue is the final goal position in the path. Red cells need to be covered manually by the operator at a later time. Typically, they appear near a wall or obstacle and cannot be reached using macro-cell discretization.

Each trial consisted of only one attempt at each cell, so some uncovered cells could be covered from a different direction on a subsequent attempt. The robot searched for contamination in Trials 1-4 and 6. All trials but Trial 3 showed that the robot detected the contamination and alerted the remote station. Trial 3 failed due to the proximity of the contamination source to the wall. The results are summarized in Table 5.1 below.

Table 5.1. CCPP results for contamination survey [71].

Trial	Contamination present	Contamination detected	Obstacles present	Completeness (%)
1	0	0	0	67.5
2	1	1	0	66.5
3	2	1	0	66.5
4	1	1	1	67.5
5	0	0	1	67.5
6	1	1	2	51.9
7	0	0	2	51.9
8	0	0	1	64.1

Several implications come from this data. First, the completeness percentage is based on a single attempt and the robot is able to cover approximately 63% on average. Poorer completeness in Trials 6 and 7 is directly related to the extra obstacle. Second, the operator has a map of what the robot did not cover. This information is necessary to prevent a completely manual re-attempt

at coverage. Finally, contamination has no effect on the robot's ability to continue coverage. As long as the operator clears it, it will keep the map in memory and continue its task.

Naturally, comparisons to commercial coverage robots such as vacuum cleaners arise. The RCTbot fails to cover areas around obstacles and near walls while a commercial vacuum cleaner robot seemingly covers the entire space. Several differences exist between typical cleaning systems and the RCTbot demonstrating the sophistication of the latter. First, commercial vacuum cleaners typically do not plan paths, instead selecting random turn and drive forward actions. The belief here is that eventually, all area will be covered. This method is not sufficient for radiation coverage due to the infinitesimally small volume of radioactive sources. For sufficiently large areas, random paths will only complete coverage in the limit of time. Next, commercial systems do not record their paths. Given that they cannot assure complete coverage, there is no way an RCT could rely on such a system. These alone condemn the use of commercial coverage robots for radiation monitoring, but one final consideration regards the difference in instrumentation. A commercial coverage robot does not possess sensitive detectors that might be damaged by collisions. As such, colliding with walls due to imprecise localization is not a necessary problem to solve. In fact, many systems rely on collisions to reset directions at least part of the time.

Overall, the CCPP algorithm described in Section 3.2 and Section 3.3 is sufficient to perform according to the requirements. It is clear that there remains room for improvement, but that is not a priority in this work.

5.3.2. Spatio-Temporal Map Maintenance

Demonstrating the capabilities of STMM is the primary objective of this research effort, and thus prioritized over improving CCPP. A major objective of this thesis is to show how spatio-temporal reasoning can improve the long-term autonomy of a robot in terms of navigation planning. To do so does not require the space to be large or necessarily involve long inspection durations. This section validates the improvement through experimental means by examining the performance of the RCTbot in a coverage survey under different spatio-temporal scenarios.

5.3.3. Experiment Design

Chapter 4 detailed the requirements and claims for STMM. This section describes an experiment to validate those claims. The experimental environment is an extension of the one shown in Figure 5-2. Now the robot must traverse the hallway-like environment shown in Figure 5-4.

Figure 5-4: Hallway environment for STMM validation experiment.

The RCTbot will make observations of the environment with its planar LIDAR as it navigates through the hallway. Each spatial scan will represent a moment in time for the BSA algorithm. Walls represent the only static objects while boxes, barrels, tunnels, and other miscellaneous objects are dynamic. The robot will observe some of the dynamic objects as static for the time scale. Figure 5-5 shows the initial environment configuration.

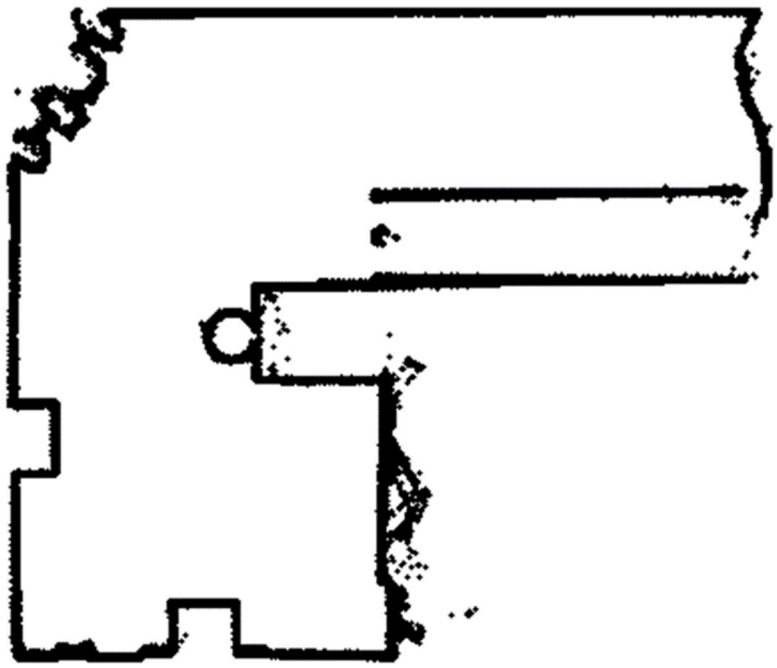

Figure 5-5: Initial static map of STMM hallway.

The initial configuration will start as a static map with probabilities for all points being near 1. Two types of dynamic objects will be added and moved for this experiment, boxes and barrels. Examples of both are shown in Figure 5-5. Table A-1 in Appendix A provides a schedule of when objects are added and removed to the environment.

The RCTbot will use the same LIDAR data from each spatial scan to compare with the static map and measure the effectiveness three different map maintenance schemes. In Scheme A, the robot performs no map maintenance and continues to use the original static map throughout time, in Scheme B, the robot performs a naïve map update which will be described below, and in Scheme C, the robot performs the BSA-based STMM described in Chapter 4 and in Section 5.2.3. The other methods serve as a comparison to demonstrate the effectiveness of BSA based STMM. For each scheme, the experiment will compare the average localization score after progressively many surveys. The localization score is a number generated with `rosarnl` based upon how many points in the static map match the observed LIDAR data within a small margin of error.

Scheme B requires more explanation. The CCPP in [71] also uses the naïve STMM method in Scheme B to help maintain its map. This method maintains a counter for each point in the map. When the robot senses occupancy at that point, it increments the counter; else, it decrements. Figure 5-6 shows the results of the naïve STMM implementation for the situation of a box moving from one corner to another.

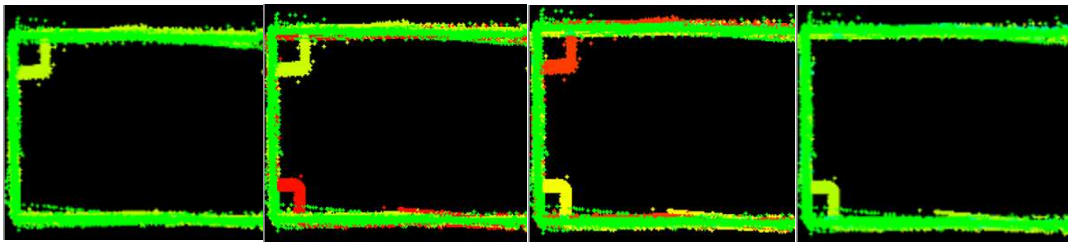

Figure 5-6: Heat map showing naïve STMM approach for a dynamic box.

The robot represents the temporal data with a series of heat maps where green means more static and red means less static. The walls have a counter much higher than the static map threshold

because they never change. The robot observes the transplanted box enough that the initial position goes from green to red and then disappears while the final position does the opposite.

5.3.4. Results

The RCTbot collected data for 100 time periods (simulating hours, days, weeks, etc.). For simplicity sake, the time periods will be taken as days for the rest of this section. Throughout the testing period, the RCTbot recorded its average localization score after each scan. The average localization is a measure of how well the robot's LIDAR measurements align with the world map at any given moment with a perfect match receiving a score of 1. As it performs coverage, the RCTbot maintains a rolling average of the score and publishes it at task completion. The average localization results are summarized in 10-day intervals in Table 5.2.

Table 5.2. Localization scores over time (percentage match).

Scheme	Day										
	1	10	20	30	40	50	60	70	80	90	100
A: Static	0.982	0.951	0.955	0.952	0.919	0.910	0.890	0.878	0.874	0.883	0.829
B: Naïve	0.967	0.967	0.968	0.925	0.921	0.970	0.956	0.938	0.960	0.927	0.914
C: STMM	0.978	0.973	0.967	0.969	0.965	0.971	0.966	0.968	0.968	0.973	0.974

It is clear that Scheme C consistently performs well while Scheme A consistently regresses and Scheme B is more erratic. To understand this performance, consider Day 30 where Scheme B and Scheme C start to diverge. Figure 5-7 shows the LIDAR data the RCTbot collected for that period.

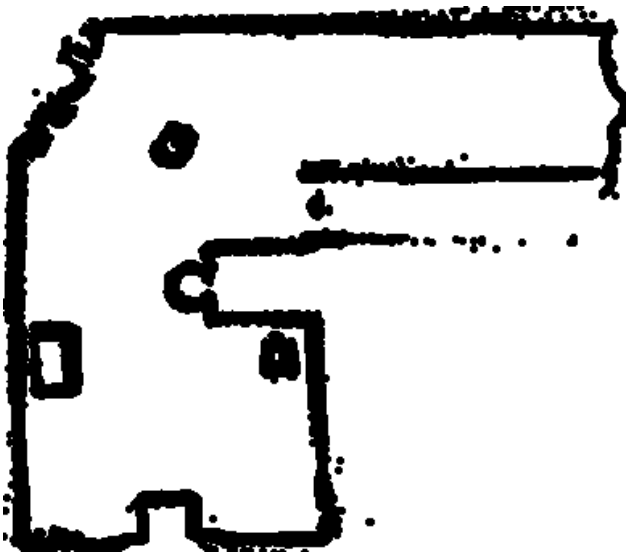

Figure 5-7: LIDAR data for Day 30.

Given the history of the obstacle locations (refer to Table A-1), the barrel in the top half is relatively the most permanent obstacle other than the walls. The box recently moved out away from the wall. Some portions will remain in the probability map, but others will not. Figure 5-8 shows the probability maps generated from the LIDAR data using Scheme B and Scheme C respectively.

Figure 5-8: Probability map for Scheme B (left) and Scheme C (right) at Day 30.

Both schemes contain the correct obstacles, but the naïve incremental method means that not all objects have reached the threshold for the static map. For a new object to become static, Scheme B requires approximately 20 days while Scheme C requires as few as 3 days. Until an object reaches the threshold, it will not appear in the static map, so the RCTbot cannot use it to localize. Figure 5-9 provides the static map generated by Scheme B and Scheme C.

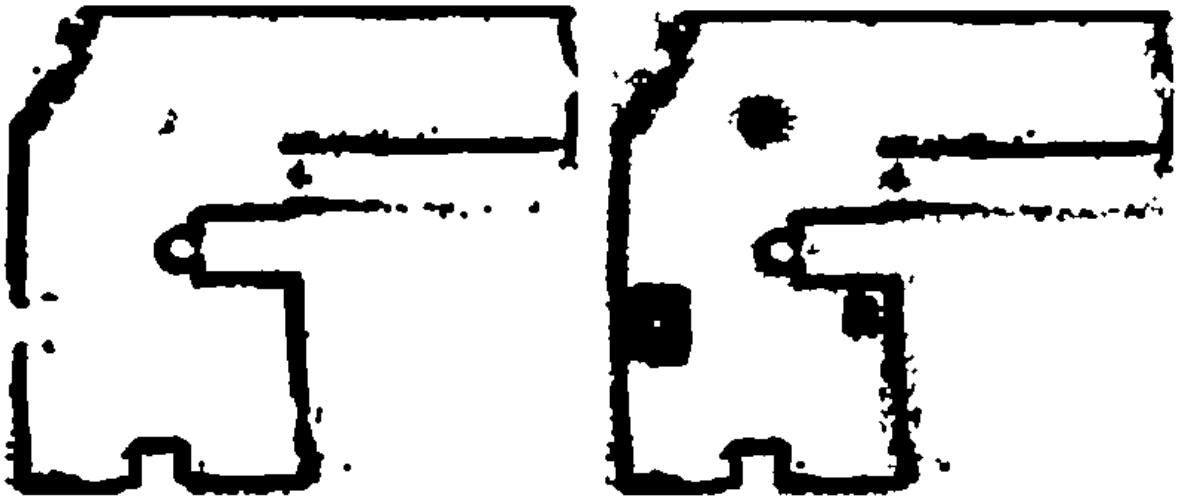

Figure 5-9: Static map for Scheme B (left) and Scheme C (right) at Day 30.

The inflexibility of Scheme B leads to a loss of localization features. After the box on the left wall was moved out slightly, Scheme B cannot fill in the wall. This causes the localization score to be worse for Scheme B in this instance than Scheme A. Scheme C, however, correctly predicts what the map looks like.

Obstacle locations, probability maps, and static maps for the other days are included in the Appendices. Figure 5-10 visualizes the data from Table.

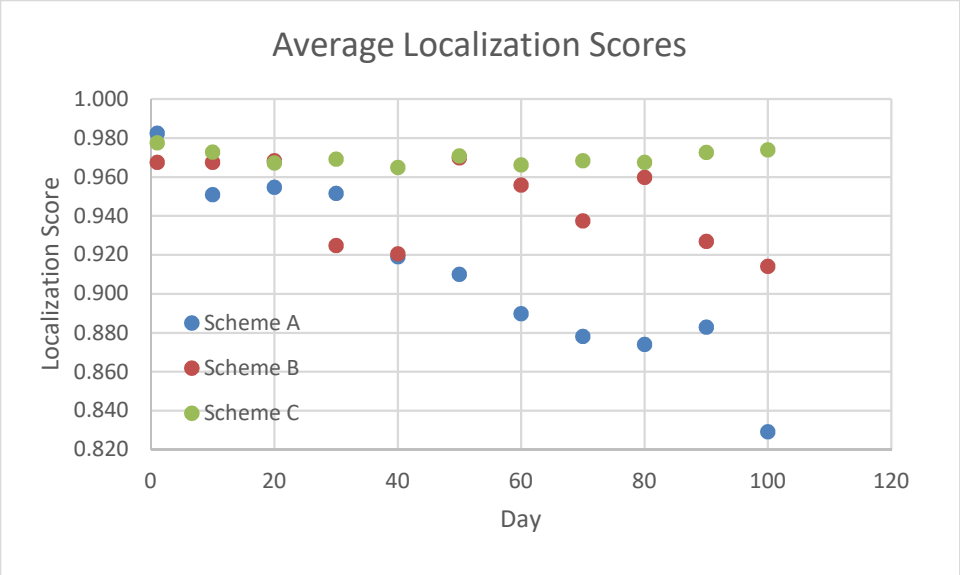

Figure 5-10: Average localization scores of the 3 map maintenance schemes.

The comparative reliability of Scheme C is clear in the data. Scheme A degrades according to expectations for a static map, becoming less reliable as time passes and more obstacles are introduced. Scheme B is unreliable due to its inadaptability as perceived changes have a smaller impact on the probability map than in Scheme C.

5.4. SUMMARY

Overall, this Chapter has shown how STMM with BSA can extend the autonomy of a mobile robot in long-term applications. Section 5.3.2 not only demonstrated the adaptability provided by the STMM algorithm, but also shows the need for some form of map maintenance. The primary result of this work is a proof of concept that static maps methods are still viable in autonomous robotics by employing map maintenance.

Chapter 6: Conclusions and Future Work

6.1. Summary of Research

Robots are increasingly needed to protect humans from dangerous working environments. Society will not rely on robots until they demonstrate the ability to maintain long-term autonomy in realistic, complex environments. This effort provides both methodology and evidence for improving long-term autonomy in semi-static environments. The RCTbot performs spatio-temporal map maintenance during radiation coverage surveys and localizes in the map to perform its task better than other approaches.

6.2. Future Work

Despite the RCTbot already demonstrating improved autonomy in long-term situations, future efforts can further enhance its abilities. Three main features immediately appear as next steps: navigation and planning with three-dimensional sensing, informative priors and more intelligent logic in the BSA algorithm, and increased mobility in the robotic platform. This section discusses these features in more detail.

6.2.1. 3-D Sensing

The current RCTbot uses a 2-D LIDAR for localization and mapping. The LIDAR senses a horizontal plane approximately 6” from the ground with a field of view of 270°. This is sufficient for navigating in a pre-made map, but it requires the operator to manually drive the robot around so that it can make this map without running into low hanging objects like tables or items close to the ground such as pallets. For safe, autonomous navigation, a robot requires a 3-D LIDAR to identify obstacles for its entire volume and not just a plane. Figure 6-1 shows a Velodyne VLP-16 3-D LIDAR [72].

Figure 6-1: Velodyne VLP-16 3-D LIDAR and control box.

This LIDAR mounted above the robot can give a 360° view of the objects surrounding it as well as 15° of vertical resolution. Figure 6-2 visualizes the data produced by the VLP-16.

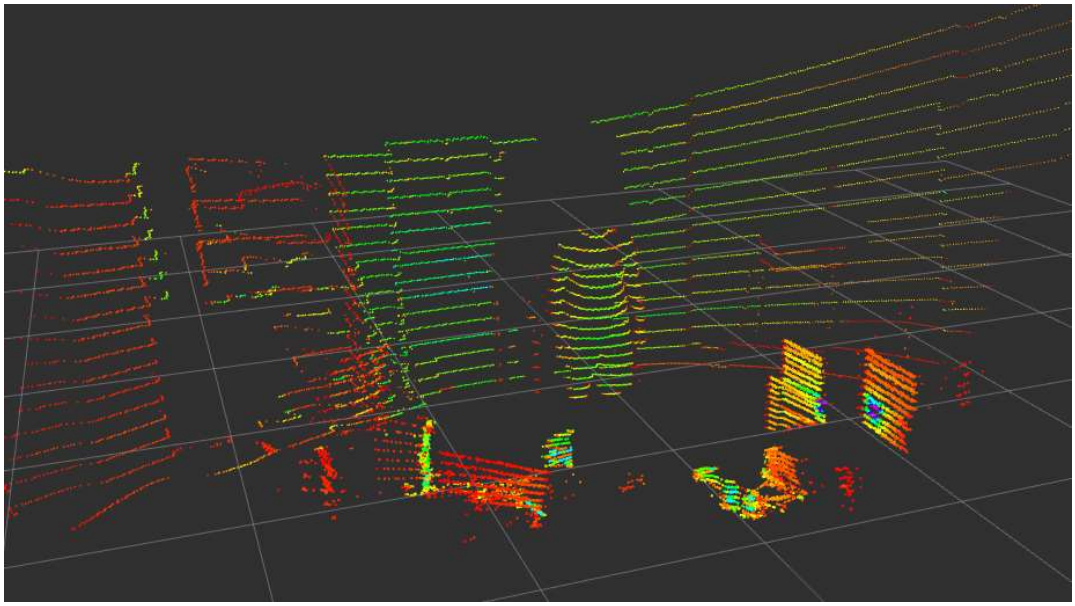

Figure 6-2: 3-D data from the VLP-16.

Current work performed in the NRG group at UT Austin shows that 3-D points can be collapsed to form 2-D obstacles in the map [73]. The 2-D obstacles yield better performance during the scan matching component of localization. With 3-D localization and mapping, the RCTbot can autonomously explore unknown environments and create new maps for surveys, thus increasing its overall self-agency.

6.2.2. Improved STMM

Some caveats about STMM exist that must eventually be addressed. Earlier chapters put emphasis on robotics in real environments. The bulk of this work was completed in the real world, yet it was a very tightly controlled environment. More needs to be done to increase the ability of the robot to work in real tasks in real environments. Researchers must continue to collect long-term viability data in a continuous evaluation in order to validate STMM. They must also explore scalability of the algorithm when faced with increased complexity and rate of change in the environment. Grid based methods have complexity based on dimension and resolution. As these values increase, STMM must process the larger amounts of data more intelligently.

Other ways the STMM algorithm presented in this work can be improved include: a) using a probability density function to describe the location of the occupied points and b) including an informative prior in the BSA algorithm based on temporal characteristics of 3-D points collected into objects.

The first method involves using a PDF such as a normal distribution to describe the location of points in the static map. This will account for minor uncertainty in the sensor distance measurements and prevent the STMM from adding too many new points to the map each time its sensor has fluctuations in the readings.

The second method again requires 3-D LIDAR data. Object recognition is a prominent topic in computer vision both in the general robotics community and in NRG. The robot can be augmented with object recognition to process 3-D point cloud and assign individual points to an object. It can then classify the objects based on temporal characteristics. A collection of points could form a barrel, which has certain temporal characteristics. These characteristics could inform the decay constant, λ_T, from Section 4.3.4 to generate an informative prior. In this manner, the robot can use knowledge about the environment to make its predictions for planning and generate better results.

6.2.3. Enhanced Robotic Platform

An enhanced robotic platform would make future efforts in CCPP more effective. In order to cover more space in a real environment, a robot requires maximum mobility to maneuver into tighter spaces. Holonomic vehicles have been popular in mobile robotics for decades, but the community has not readily adopted them due to the difficulty of controlling them and determining the uncertainty in odometry measurements. As mentioned in Section 5.2.1, ROS provides drivers for a variety of systems, including holonomic robots, making it more convenient for research to utilize their unique characteristics.

A holonomic mobile robot can move in any direction despite its initial state. Returning to the discretization discussion of Section 3.3.1, micro-cells enable a robot to more precisely cover the space, but traditional mobile platforms with differential steering struggle to plan paths to cover micro-cells sequentially in unexpected situations. A holonomic system has no such difficulties. Figure 6-3 shows a developmental holonomic robot from the NRG.

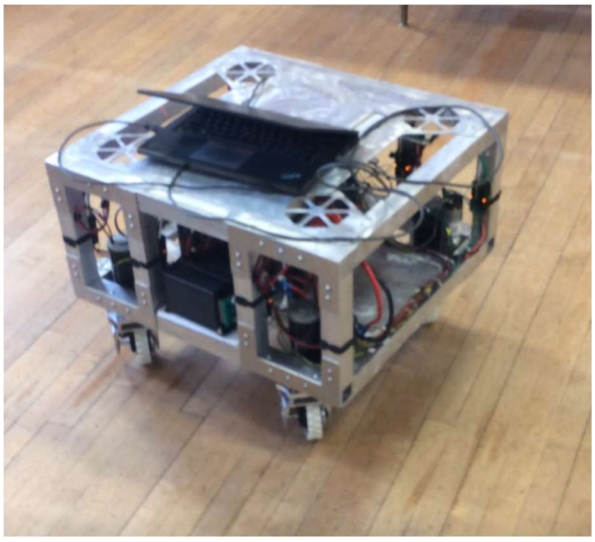

Figure 6-3: The NRG's holonomic platform with four actuated-caster swerve steer wheels.

This platform has four wheels with two motors each. The first motor turns the wheel as normal, and the second one rotates the wheel like an actuated caster. Upon mounting a radiation sensor to this platform, a robot could cover the entire space without ever putting its wheels before its sensor. This means that there is no chance that the robot would contaminate its wheels and spread that contamination. This enables the robot to sense radiation, mark it, and continue its survey with no fear of tracking contamination. Given this, an operator only has to check the locations after the robot has finished, and there is no need for cleaning and resetting the robot. This step of increased autonomy is needed before a robotic system will be truly viable in a real radiation environment.

6.3. Alternative Applications

The work in this thesis was focused on the single application of radiation contamination coverage surveys. Aspects of the work have broader applications for STMM. CCPP algorithms often show utility in areas such as cleaning, humanitarian demining, agriculture, and surveillance. An algorithm such as STMM or one like it is necessary for CCPP to be possible in realistic environments. STMM is not useful only for CCPP algorithms though. Any navigation problem

based on localization in a map by sensor scan matching can only be improved by the addition of spatio-temporal reasoning. If a map is used, map maintenance is necessary for long-term applications. STMM stands as a simple, hardware agnostic algorithm for robots with correlative data sensors.

6.4. CONCLUDING REMARKS

As stated in Section 1.4, the objective of this work is to show that the RCTbot is deployment ready for radiation coverage tasks. This objective serves the overarching goal of addressing the societal and technical barriers that autonomous robots must overcome for utilization in realistic scenarios. This thesis has shown that by implementing CCPP and STMM methods, the RCTbot is discernably better for radiation contamination surveys than a human RCT. There is still work to be done to improve its overall effectiveness and increase its autonomy.

Appendix A: STMM Obstacle Schedule

Table A.1. Dynamic obstacle spatio-temporal schedule.

Day	Boxes	Barrels
1	(254, 2970), (2663, 487)	--
2	(254, 2970), (2663, 487)	(2971, 6218)
3	(254, 2970), (2663, 487)	(2971, 6218)
4	(254, 2970), (2663, 487)	(2971, 6218)
5	(254, 2970), (2663, 487)	(2971, 6218)
6	(254, 2970), (2663, 487)	(2971, 6218)
7	(254, 2970), (2663, 487)	(2971, 6218)
8	(254, 2970), (2663, 487)	(2971, 6218)
9	(254, 2970), (2663, 487)	(2971, 6218)
10	(254, 2970), (2663, 487)	(2971, 6218)
11	(254, 2970), (2663, 487)	(2971, 6218)
12	(254, 2970), (2663, 487)	(2971, 6218)
13	(254, 2970), (2663, 487)	(2971, 6218)
14	(254, 2970), (2663, 487)	(2971, 6218)
15	(254, 2970), (2663, 487)	(2971, 6218)
16	(254, 2970), (2663, 487)	(2971, 6218)
17	(254, 2970), (2663, 487)	(2971, 6218)
18	(254, 2970), (2663, 487)	(2971, 6218)
19	(254, 2970), (2663, 487)	(2971, 6218)
20	(254, 2970), (2663, 487)	(2971, 6218)
21	(254, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
22	(254, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
23	(254, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
24	(254, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
25	(520, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
26	(520, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
27	(520, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
28	(520, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
29	(520, 2970), (2663, 487)	(2971, 6218), (4536, 3449)
30	(520, 2970), (2663, 487)	(2971, 6218), (4536, 3449)

31	(520, 2789), (2663, 487)	(1911, 8240), (4536, 3449)
32	(520, 2789), (2663, 487)	(1911, 8240), (4536, 3449)
33	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
34	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
35	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
36	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
37	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
38	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
39	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
40	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
41	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449)
42	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
43	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
44	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
45	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
46	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
47	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
48	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
49	(4687, 662), (2663, 487)	(1911, 8240), (4536, 3449), (9510, 6539)
50	(4687, 662), (2663, 487)	(1911, 8240), (2929, 3534), (9510, 6539)
51	(4687, 662), (2663, 487)	(1911, 8240), (2929, 3534), (9510, 6539)
52	(4687, 662), (2663, 487)	(1911, 8240), (2929, 3534), (9510, 6539)
53	(4687, 662), (2663, 487)	(1911, 8240), (2929, 3534), (9510, 6539)
54	(4687, 662), (2663, 487)	(1911, 8240), (2929, 3534), (9510, 6539)
55	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (9510, 6539)
56	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (9510, 6539)
57	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (9510, 6539)
58	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (9510, 6539)
59	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (2244, 8064)
60	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (2244, 8064)
61	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (2244, 8064)
62	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (2244, 8064)
63	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (2244, 8064)
64	(4687, 662), (2663, 487)	(1911, 8240), (4347, 5320), (2244, 8064)
65	(4687, 662), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
66	(4687, 662), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
67	(4687, 662), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)

68	(4687, 662), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
69	(4687, 662), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
70	(4337, 529), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
71	(4337, 529), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
72	(4337, 529), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
73	(4337, 529), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
74	(4337, 529), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
75	(4337, 529), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
76	(4672, 659), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
77	(4672, 659), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
78	(4672, 659), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
79	(4672, 659), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
80	(4672, 659), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
81	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
82	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
83	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
84	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
85	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
86	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
87	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
88	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
89	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
90	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
91	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
92	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
93	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
94	(4331, 532), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
95	(4252, 5585), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
96	(4252, 5585), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
97	(4252, 5585), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064)
98	(4252, 5585), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064), (3818, 624), (4461, 634)
99	(4252, 5585), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064), (3818, 624), (4461, 634)
100	(4252, 5585), (2663, 487)	(1911, 8240), (3093, 8091), (2244, 8064), (3818, 624), (4461, 634)

Appendix B: LIDAR Data

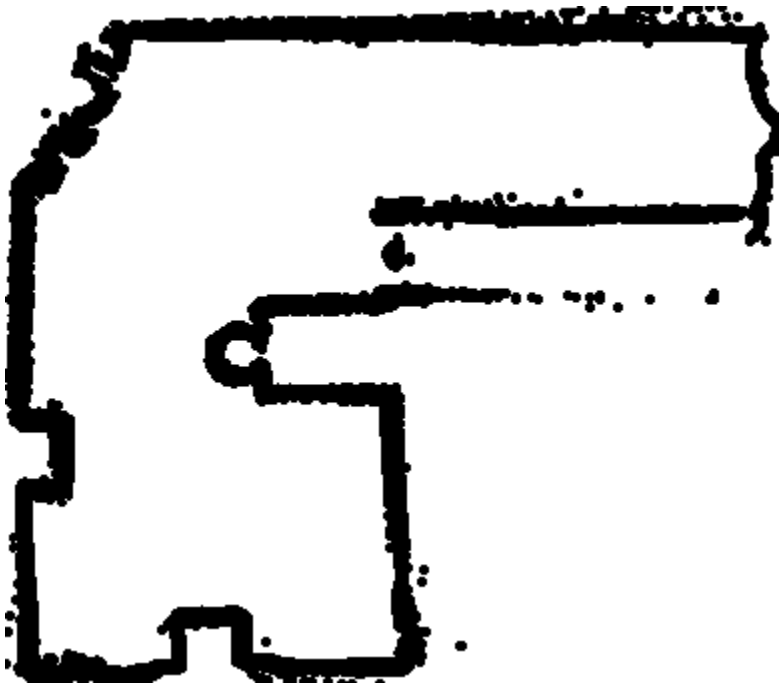

Figure B-1: Day 1

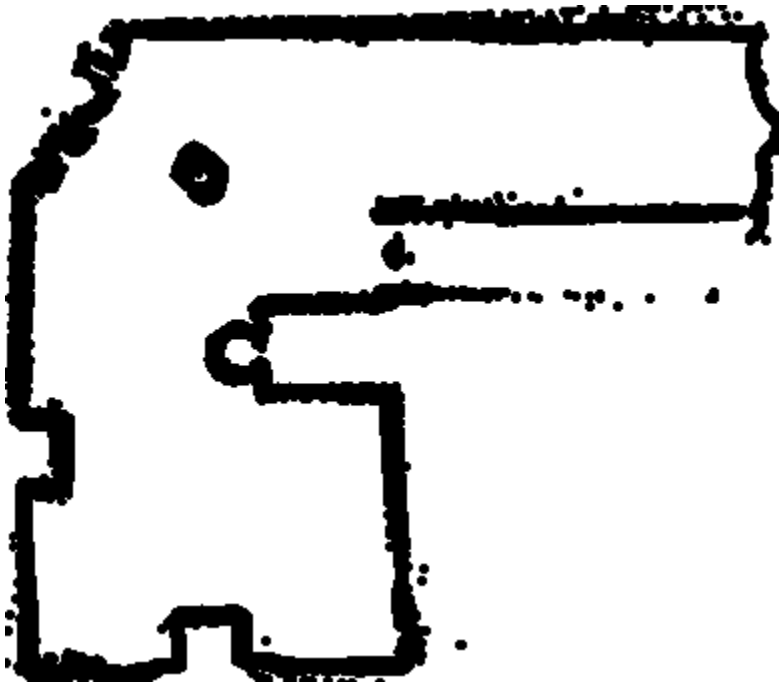

Figure B-2: Day 10

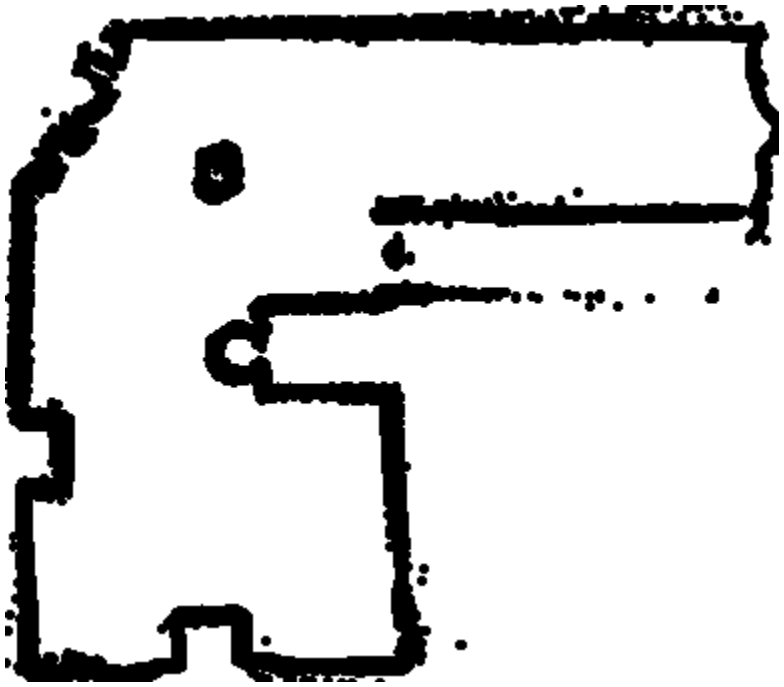

Figure B-3: Day 20

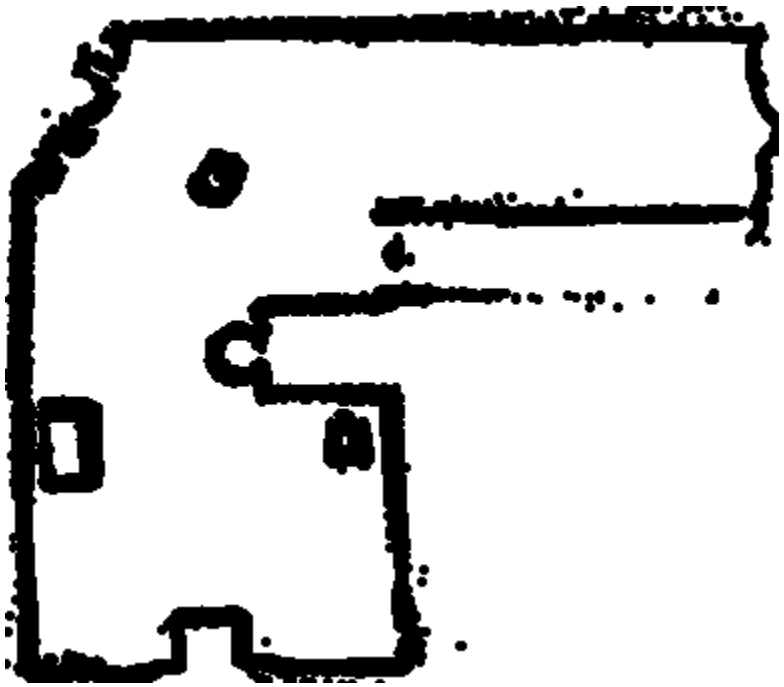

Figure B-4: Day 30

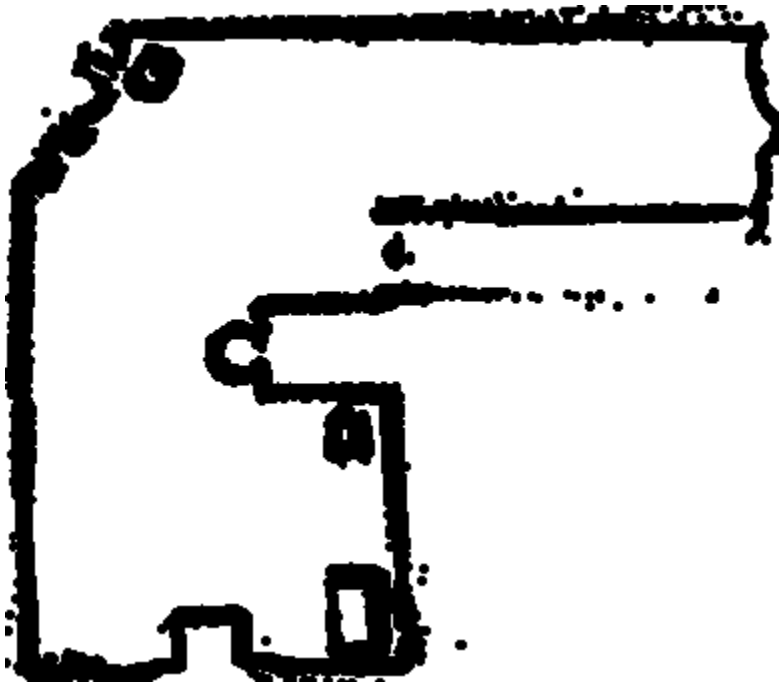

Figure B-5: Day 40

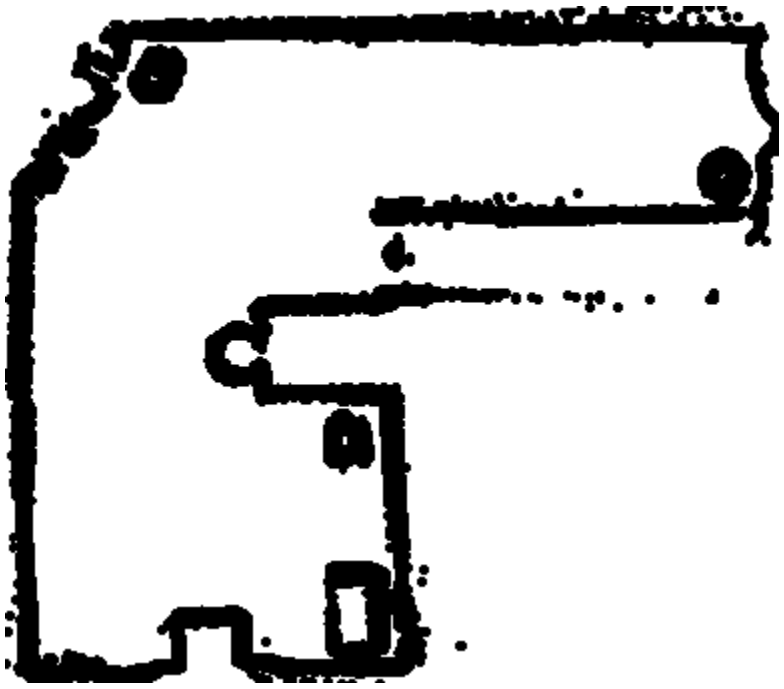

Figure B-6: Day 50

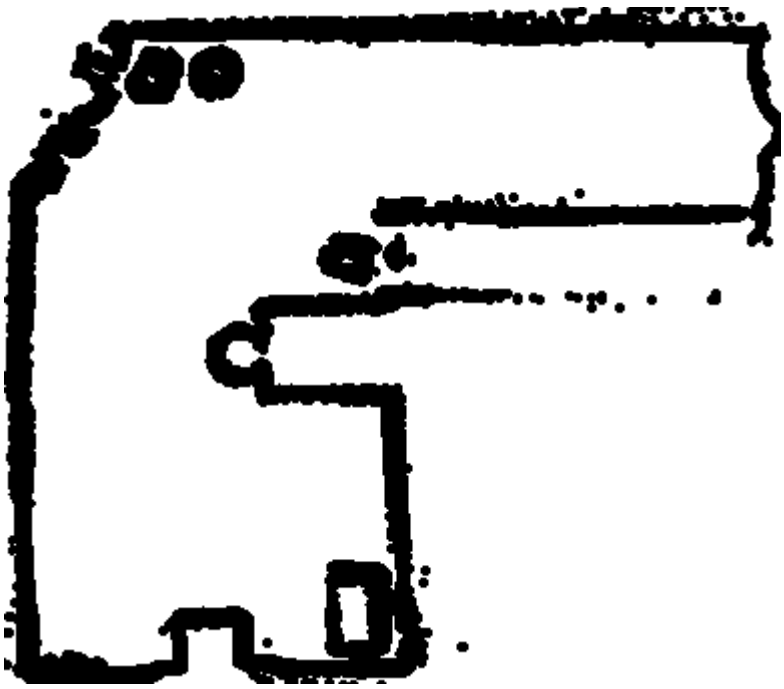

Figure B-7: Day 60

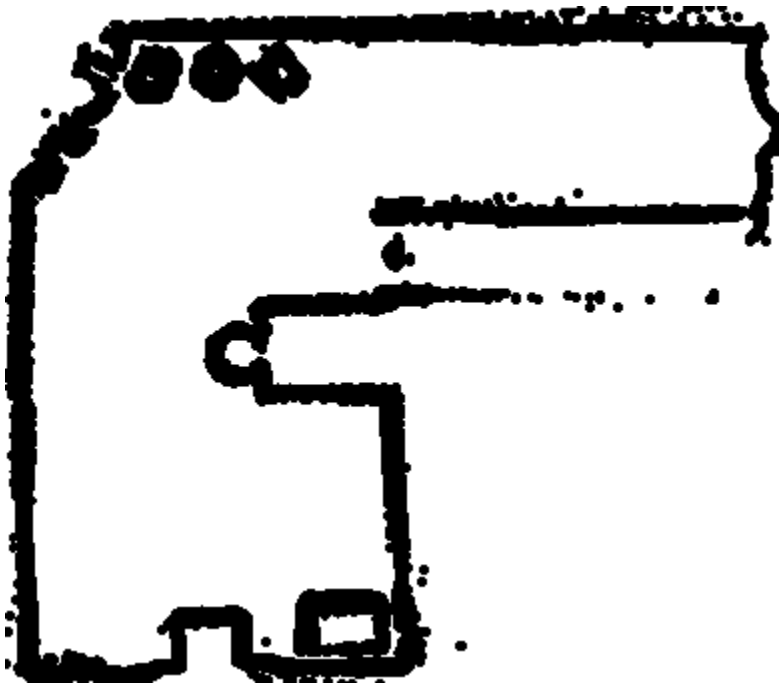

Figure B-8: Day 70

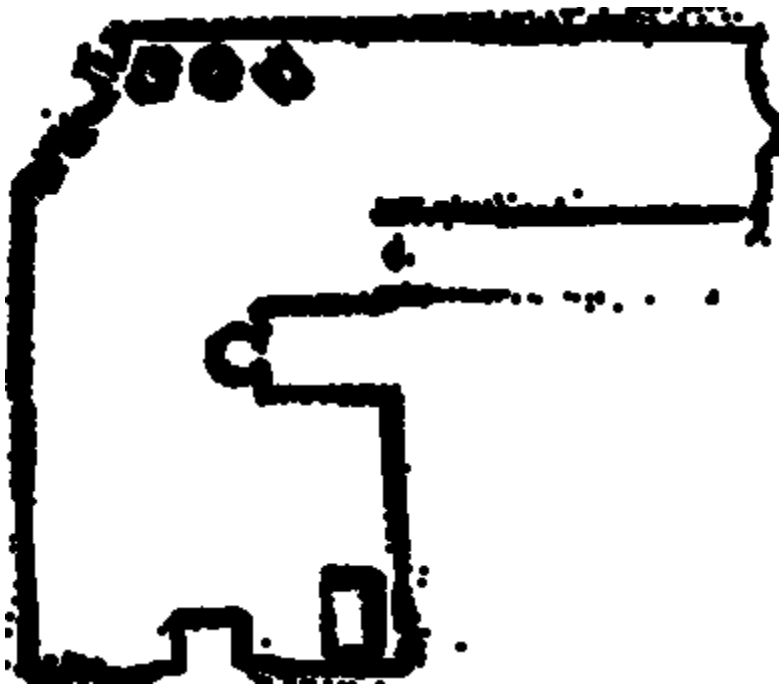

Figure B-9: Day 80

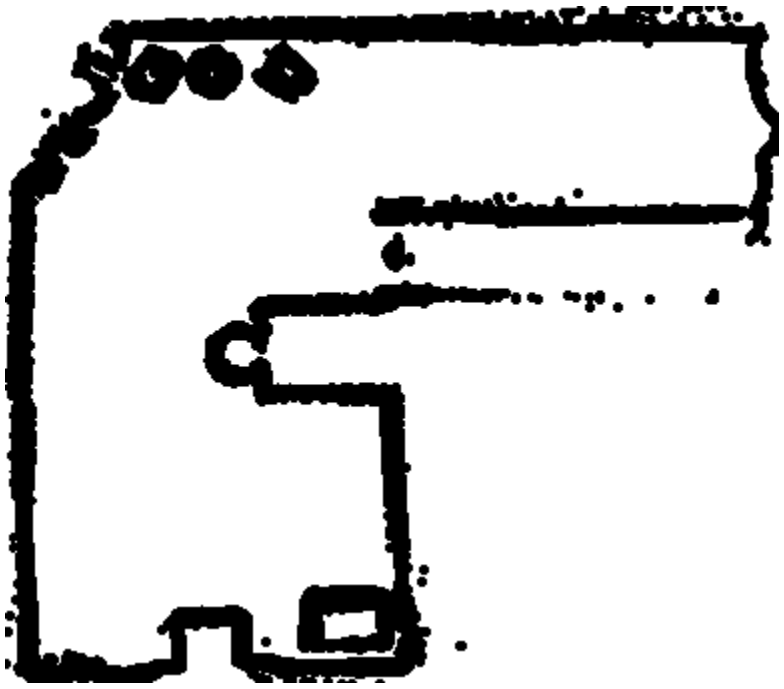

Figure B-10: Day 90

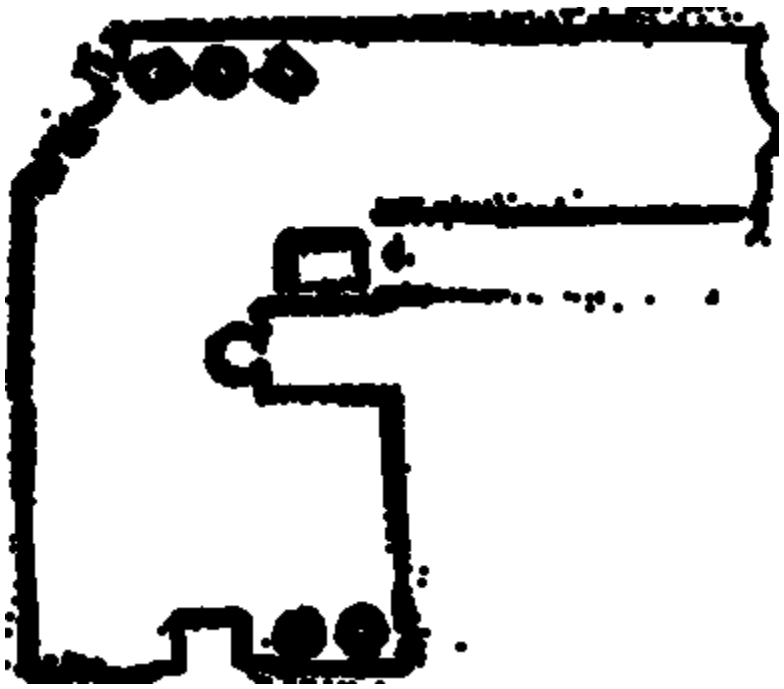

Figure B-11: Day 100

Appendix C: Probability Maps

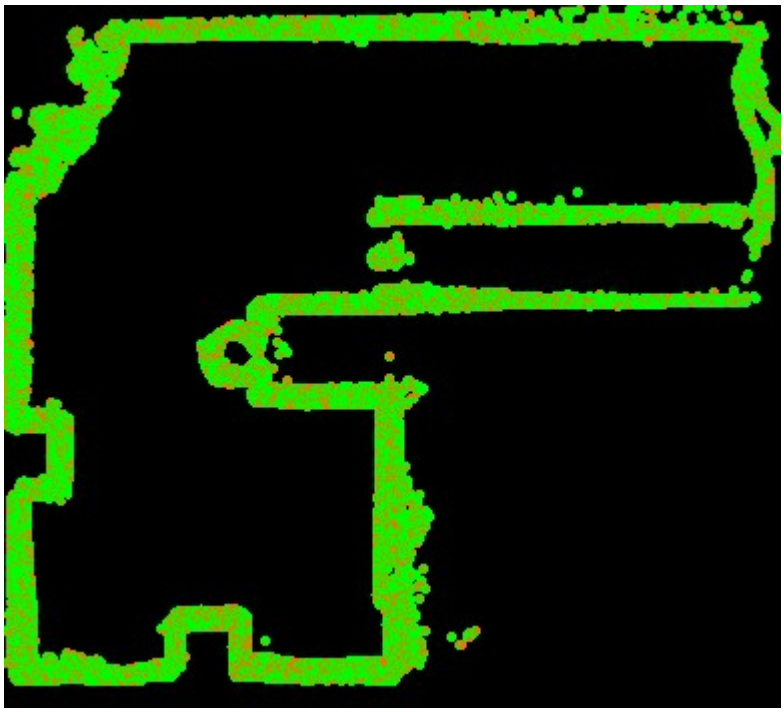

Figure C-1: Day 1

Figure C-2: Day 1

Figure C-3: Day 1

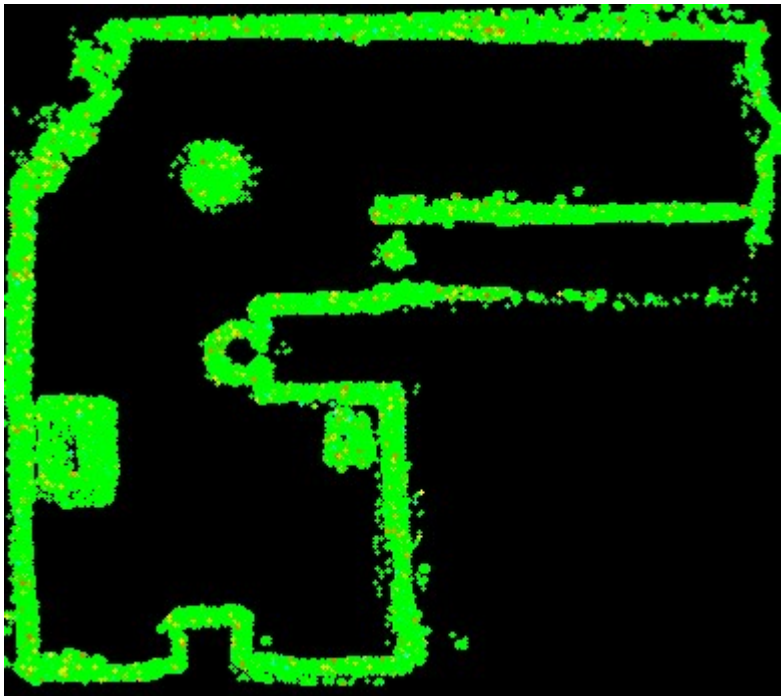

Figure C-4: Day 1

Figure C-5: Day 40

Figure C-6: Day 50

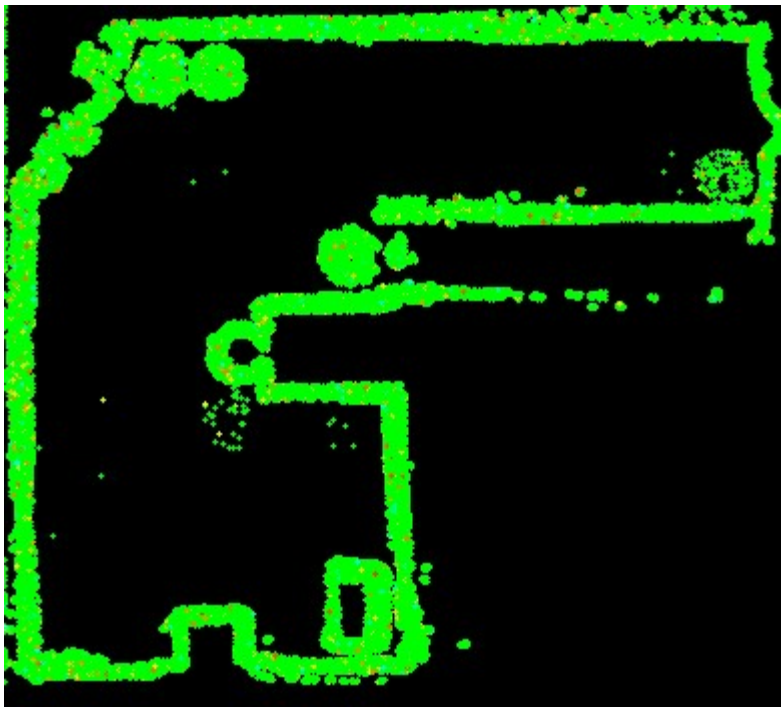

Figure C-7: Day 60

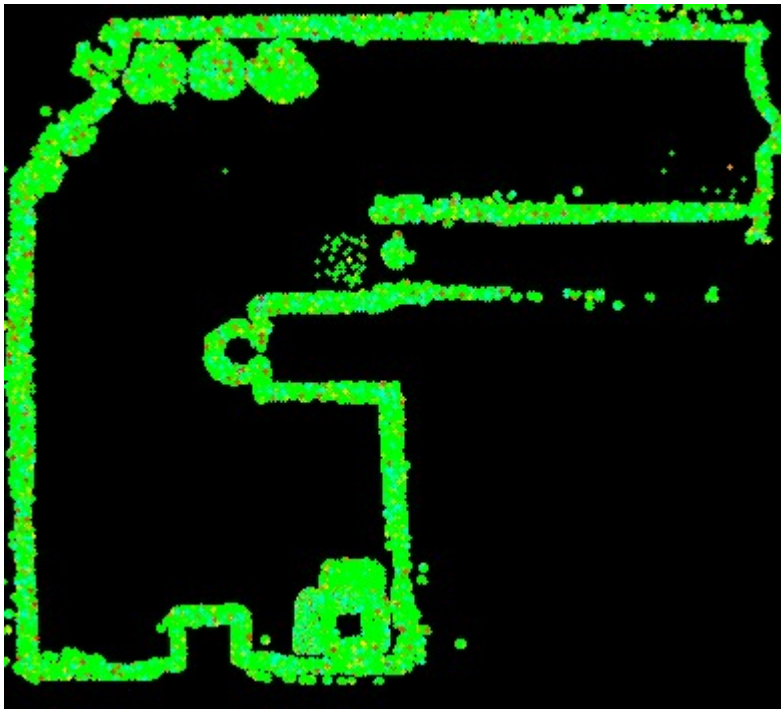

Figure C-8: Day 70

Figure C-9: Day 80

Figure C-10: Day 90

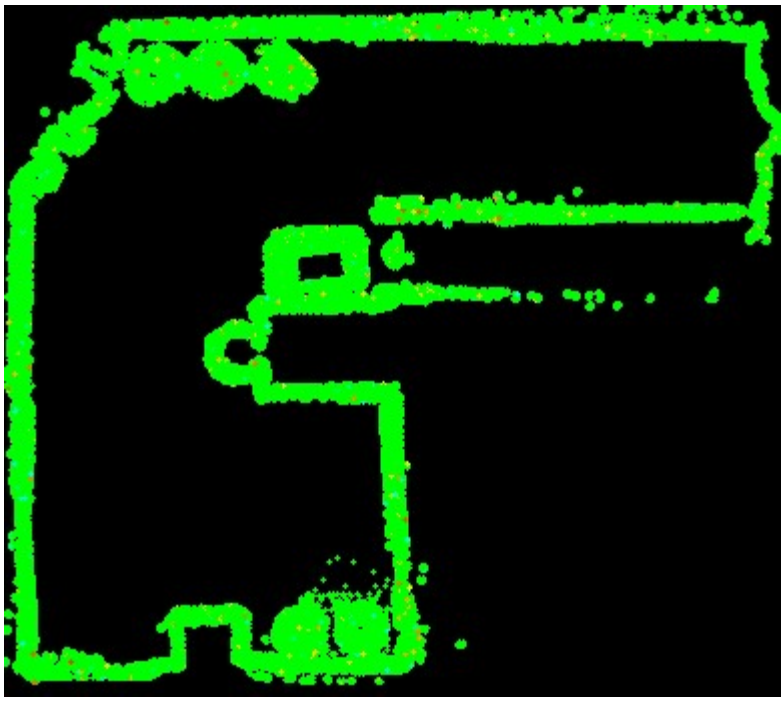

Figure C-11: Day 100

References

[1] J. M. Beer, A. D. Fisk and W. A. Rogers, "Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction," *Journal of Human-Robot Interaction,* vol. 3, no. 2, p. 74, 1 6 2014.

[2] C. Westin, C. Borst and B. Hilburn, "Strategic Conformance: Overcoming Acceptance Issues of Decision Aiding Automation?," *IEEE Transactions on Human-Machine Systems,* vol. 46, no. 1, pp. 41-52, 2 2016.

[3] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce and H. P. Beck, "The role of trust in automation reliance," *International Journal of Human-Computer Studies,* vol. 58, no. 6, pp. 697-718, 6 2003.

[4] R. Parasuraman and C. D. Wickens, "Humans: Still Vital After All These Years of Automation," *Human Factors,* vol. 50, no. 3, pp. 511-520, 6 2008.

[5] M. Harbers, M. M. M. Peeters and M. A. Neerincx, "Perceived Autonomy of Robots: Effects of Appearance and Context," Springer International Publishing, 2017, pp. 19-33.

[6] *Occupational Dose Limits,* 1991.

[7] Department of Energy, *Guide of Good Practices for Occupational Radiological Protection in Plutonium Facilities,* 2008.

[8] Los Alamos National Laboratory, "P121, Radiation Protection," Los Alamos, 2016.

[9] H. Cember and T. E. (. E. Johnson, Introduction to health physics, McGraw-Hill Medical, 2009.

[10] G. L. Voelz and I. G. Buican, "Plutonium and Health," *Los Alamos Science,* vol. 26, pp. 74-89, 2000.

[11] W. Moss and R. Eckhardt, "The Human Plutonium Injection Experiments," *Los Alamos Science,* vol. 23, pp. 177-233, 1995.

[12] Department of Energy, *Radiological Control,* 2008.

[13] T. DeRoma, "LANL defends plutonium facility after critical report," *The Los Alamos Monitor,* 23 6 2017.

[14] S. Thrun, W. Burgard and D. Fox, PROBABILISTIC ROBOTICS, 1999.

[15] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems,* vol. 61, pp. 1258-1276, 2013.

[16] H. Choset, "Coverage for robotics–A survey of recent results," *Annals of mathematics and artificial intelligence,* pp. 113-126, 2001.

[17] R. R. Fuchs and S. A. Costigan, "RP-SOP-037 Surveying for Fixed and Removable Contamination," Los Alamos, 2016.

[18] "RP-1-DP-48," Los Alamos, 2015.

[19] G. F. Knoll, Radiation detection and measurement, John Wiley, 2010, p. 830.

[20] J. Durham, M. Johnson and D. Gardner, "Contamination surveys for release of material," Richland, WA, 1994.

[21] L. H. Munson, C. W. N Herrington and D. P. Higby R L Kathren, "Health Physics Manual of Good Practices for Reducing Radiation Exposure to Levels that are As Low As Reasonably Achievable (ALARA)," 1988.

[22] R. Kathren and R.L., "Guide to reducing radiation exposure to as low as reasonably achievable (ALARA)," Richland, WA (United States), 1980.

[23] N. Tsoulfanidis and S. Landsberger, Measurement and Detection of Radiation., 4th ed., CRC Press, 2015, p. 606.

[24] A. V. Klimenko, W. C. Priedhorsky, N. W. Hengartner and K. N. Borozdin, "Efficient strategies for low-statistics nuclear searches," *IEEE Transactions on Nuclear Science,* 2006.

[25] A. Kumar, H. G. Tanner, A. V. Klimenko, K. Borozdin and W. C. Priedhorsky, "Automated sequential search for weak radiation sources," in *14th Mediterranean Conference on Control and Automation, MED'06*, 2006.

[26] R. A. Cortez, X. Papageorgiou, H. G. Tanner, A. V. Klimenko, K. N. Borozdin and W. C. Priedhorsky, "Experimental implementation of robotic sequential nuclear search," in *2007 Mediterranean Conference on Control and Automation, MED*, 2007.

[27] R. B. Anderson, "Development of Mobile Platform for Inventory and Inspection Applications in Nuclear Environments," 2015.

[28] B. Ebersole, "Skid-Steer Kinematics for Dual-Arm Mobile Manipulator System with Dynamic Center of Gravity," 2016.

[29] Z. L. Cao, Y. Huang and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *Journal of Robotic Systems,* vol. 5, no. 2, pp. 87-102, 4 1988.

[30] A. Zelinsky, R. Jarvis, J. C. Byrne and S. Yuta, "Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot," *IN PROCEEDINGS*

OF INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, vol. 13, pp. 533--538, 1993.

[31] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*.

[32] E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *Discrete Applied Mathematics,* vol. 55, no. 3, pp. 197-218, 12 1994.

[33] D. W. Gage, "Randomized search strategies with imperfect sensors," in *Proceedings of SPIE Mobile Robots VIII*, Boston, 1993.

[34] T. Balch, "The Case for Randomized Search," in *In Workshop on Sensors and Motion, IEEE International Conference on Robotics and Automation*, San Francisco, 2000.

[35] J.-C. Latombe, Robot motion planning, Kluwer Academic Publishers, 1991, p. 651.

[36] P.-M. Hsu, C.-L. Lin, M.-Y. Yang, P.-M. Hsu, ·. C.-L. Lin and M.-Y. Yang, "On the Complete Coverage Path Planning for Mobile Robots," *J Intell Robot Syst,* vol. 74, pp. 945-963, 2014.

[37] R. Yehoshua, N. Agmon and G. Kaminka, "Robotic adversarial coverage of known environments," *The International Journal of Robotics Research,* vol. 35, no. 12, pp. 1-26, 2016.

[38] T.-K. K. Lee, S.-H. H. Baek, Y.-H. H. Choi and S.-Y. Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Systems,* vol. 59, no. 10, pp. 801-812, 2011.

[39] G. P. Strimel and M. M. Veloso, "Coverage planning with finite resources," in *IEEE International Conference on Intelligent Robots and Systems*, 2014.

[40] R. Yehoshua, N. Agmon and G. A. Kaminka, "Safest path adversarial coverage," in *IEEE International Conference on Intelligent Robots and Systems*, 2014.

[41] T. Bretl and S. Hutchinson, "Robust coverage by a mobile robot of a planar workspace," in *2013 IEEE International Conference on Robotics and Automation*, 2013.

[42] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar and D. Hull, "Morse Decompositions for Coverage Tasks," *The International Journal of Robotics Research,* vol. 21, no. 4, pp. 331-344, 2 4 2002.

[43] E. U. Acar and H. Choset, "Sensor-based Coverage of Unknown Environments: Incremental Construction of Morse Decompositions".

[44] 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,* vol. 22, no. 7-8, pp. 441-466, 2 7 2003.

[45] S. C. Wong and B. A. MacDonald, "Complete Coverage by Mobile Robots Using Slice Decomposition Based on Natural Landmarks," Springer, Berlin, Heidelberg, 2004, pp. 683-692.

[46] S. Wong, "Qualitative Topological Coverage of Unknown Environments by Mobile Robots," 2006.

[47] Chaomin Luo, S. Yang, D. Stacey and J. Jofriet, "A solution to vicinity problem of obstacles in complete coverage path planning," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*.

[48] S. Yang and C. Luo, "A Neural Network Approach to Complete Coverage Path Planning," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics),* vol. 34, no. 1, pp. 718-724, 2 2004.

[49] Y. Gabriely and E. Rimon, "Spiral-STC: an on-line coverage algorithm of grid environments by a mobile robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*.

[50] R. Biswas, B. Limketkai, S. Sanner and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* vol. 1, pp. 1014-1019, 2002.

[51] C. Stachniss and W. Burgard, "Mobile robot mapping and localization in non-static environments," *Proceedings of the 20th national conference on Artificial intelligence - Volume 3,* pp. 1324-1329, 2005.

[52] P. Biber and T. Duckett, "Dynamic Maps for Long-Term Operation of Mobile Service Robots," *Proceedings of the Robotics: Science & Systems Conference,* p. 17–24, 2005.

[53] P. Biber and T. Ducket, "Experimental Analysis of Sample-Based Maps for Long-Term SLAM," *International Journal of Robotics Research,* vol. 28, no. 1, pp. 20-33, 2009.

[54] L. Montesano, J. Minguez and L. Montano, "Modeling the static and the dynamic parts of the environment to improve sensor-based navigation," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2005.

[55] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[56] A. Walcott-Bryant, M. Kaess, H. Johannsson and J. J. Leonard, "Dynamic pose graph SLAM: Long-term mapping in low dynamic environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[57] W. Churchill and P. Newman, "Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation," in *2012 IEEE International Conference on Robotics and Automation*, 2012.

[58] T. Krajnik, J. P. Fentanes, G. Cielniak, C. Dondrup and T. Duckett, "Spectral analysis for long-term robotic mapping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[59] D. M. Rosen, J. Mason and J. J. Leonard, "Towards lifelong feature-based mapping in semi-static environments," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2016.

[60] H. Johannsson, M. Kaess, M. Fallon and J. J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *2013 IEEE International Conference on Robotics and Automation*, 2013.

[61] *Occupational Radiation Protection,* 1991.

[62] R. R. Fuchs and S. A. Costigan, "RP-SOP-031 Planning Routine Radiological Monitoring," Los Alamos, 2017.

[63] J. G. Archer and S. A. Costigan, "RP-1-DP-51," Los Alamos, 2011.

[64] K. Courville and S. A. Costigan, "RP-SOP-020 Contamination Monitoring Standard," Los Alamos, 2016.

[65] S. C. Wong, L. Middleton and B. A. MacDonald, "Performance Metrics for Robot Coverage Tasks," *IN PROCEEDINGS AUSTRALASIAN CONFERENCE ON ROBOTICS AND AUTOMATION (ACRA,* pp. 7--12, 2002.

[66] S. Behnke, "Local Multiresolution Path Planning," 2004.

[67] J. G. Ibrahim, M.-H. Chen and D. Sinha, "Bayesian Survival Analysis," in *Springer Series in Statistics*, New York, Springer-Verlag New York, Inc., 2001.

[68] K.-R. Koch, "Introduction to Bayesian Statistics".

[69] Adept MobileRobots, "Pioneer LX Mobile Research Platform," [Online]. Available: http://www.mobilerobots.com/PDFs/Pioneer%20LX%20datasheet.pdf.

[70] M. Quigley, B. Gerkey and W. D. Smart, Programming robots with ROS, 1st ed., O'Reilly Media, 2015.

[71] M. Pitsch, M. Pryor, Z. Dewey and R. Anderson, "Self-Navigating Robotic Assistant for Long-term Wide Area Floor Contamination Monitoring-18213," in *Waste Management Symposia*, Phoenix, 2018.

[72] "Velodyne LiDAR PUCK ™".

[73] C. Suarez, C. McMahon, J. Plummer, W. Wells, J. Benitez and M. Losada, "Savannah River Site H-Canyon Tunnel Inspection LiDAR Mapping Solution," in *Waste Management Symposia (Accepted)*, Phoenix, 2019.