

Mobile Robot for Retail Inventory Using RFID

by

Jian Zhang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 10, 2016

Keywords: autonomous robot, RFID, RFID based inventory, RFID tag, Localization

Copyright 2016 by Jian Zhang

Approved by

Thaddeus Roppel, Chair, Associate Professor of Electrical and Computer Engineering
John Hung, Professor of Electrical and Computer Engineering
Shiwen Mao, Samuel Ginn Endowed Professor of Electrical and Computer Engineering
Robert Dean, Professor of Electrical and Computer Engineering

Abstract

The use of RFID technology is increasing in retail stores, because it improves the performance of automated checkout, inventory and theft detection systems. In this dissertation, we present a novel implementation of a mobile robot that can perform retail inventory autonomously. The robot builds the map by its sensors for surrounding environments. With the built map, the robot can autonomously generate a path to cover all the target spaces in the environment, and then it can perform the RFID based inventory. Experimental results show that our robot can efficiently perform RFID based inventory in a retail environment with complex layout, and provides inventory accuracy that compares favorably to manual inventory.

In this dissertation, we also present two RFID tag localization algorithms, the fixed power and variable power tag localization algorithms. Both algorithms are probabilistic in nature. For fixed-power localization, the robot collects RFID tag responses at a fixed power of RFID reader while it navigates in the store. After collecting enough responses for a RFID tag in different positions, the fixed power tag localization algorithm can estimate the location for this tag by recursive Bayes updating. The variable power tag localization algorithm needs the robot to collect the RFID tag responses at varying power levels of RFID reader from multiple locations. When a tag can be read at a high power level, but cannot be read at a low power level, we can determine relative range of the tag to the robot with higher resolution than a reader working at a fixed power level. After a number of successful tag responses are measured, the variable power tag localization algorithm calculates locations for each RFID tag. The

experimental results show that the fixed power tag localization algorithm takes about 30 minutes to estimate all the location of items in our mock retail store, in contrast, the variable method needs about 6 hours to complete the same task. The variable power tag localization algorithm provides better localization accuracy, the average error is about 0.5 meter. The fixed power tag localization algorithm provides about 1.5 meter localization accuracy. Therefore, two algorithms are fitted for two different application scenarios: The fix power tag localization algorithm can be used to quickly estimate the location of merchandise, and the variable power tag localization algorithm could be deployed for a scenario where more precise localization result is needed.

Acknowledgments

First of all I would like to thank my advisor Dr. Thaddeus Roppel for his support and guidance during my time at Auburn University. I would like to thank Justin Patton, for giving me the chance to become involved with the RFID community, and providing me with all the resources for my robotic research. I would like to thank Dr. Senthilkumar CP for his guidance on RFID technology and helping me with problems regarding RFID. I would like to thank Dr. John Y. Hung, Dr. Shiwen Mao and Dr. Robert N. Dean for serving on my dissertation committee. I would also like to thank Dr. Lloyd Riggs for serving as the University Reader.

A very sincere thank you goes to my wife for her words of encouragement, willingness to listen, and constant support throughout my life. This dissertation could not have been done without her help and support.

And finally I would like to extend thanks to all my friends and colleagues at our lab for the pleasant atmosphere. Everyone has been open-minded to discussions that contributed greatly to this dissertation.

Table of Contents

Abstract.....	ii
Acknowledgments.....	iv
List of Figures.....	viii
List of Tables	x
List of Abbreviations	xi
Chapter 1 Introduction	1
1.1 Goals	3
1.2 Motivation.....	4
Chapter 2 Literature Survey.....	7
2.1 Motion Planning of Autonomous Mobile Robots.....	7
2.2 Wireless and RFID Localization Algorithms	10
Chapter 3 RFID Technology Overview	14
3.1 History of RFID	15
3.2 Architecture of a RFID System and Type of RFID	19
3.3 Technical Basics for Passive RFID.....	22
3.4 Characteristics of UHF Passive RFID	25
3.4.1 Detectable Range of UHF Passive RFID.....	25
3.4.2 Multipath Fading of UHF Passive RFID system	26
Chapter 4 Robot Navigation for Retail Environment	28

4.1 Mapping	29
4.1.1 RBPF for SLAM	31
4.1.2 Multiple-layer Mapping Algorithm with Both LIDAR and Kinect	32
4.2 Robot Localization	35
4.3 Inventory Based Path Planner	35
4.3.1 Global Path Planner	36
4.3.2 Local Path Planner	39
Chapter 5 RFID Tag Localization	43
5.1 Bayesian Updating	44
5.2 Antenna Model	46
5.3 Fixed Power Tag Localization Algorithm	48
5.4 Variable Power Tag Localization Algorithm	51
Chapter 6 Experimental Setup and Results	57
6.1 Experimental Setup	57
6.1.1 Experiment Environment	57
6.1.2 Robot System	59
6.2 Mapping Results	61
6.3 Navigational Results	64
6.3.1 Global Path Generating	64
6.3.2 Robot Navigates in the Store	67
6.4 RFID Inventory Results	68
6.5 RFID Tag Localization Results	73
Chapter 7 Conclusion and Future Work	80

7.1 Summary	80
7.2 Future Work	81
References	83

List of Figures

Figure 2.1. Positioning based on TOA.....	11
Figure 2.2. Positioning based on AOA.	12
Figure 3.1. A sheep with a RFID tag, the tag is marked by a red rectangle	17
Figure 3.2. A typical RFID system	19
Figure 3.3. Typical frequency spectrum available for RFID	20
Figure 3.4. Passive, semi-passive and active RFID system.....	21
Figure 3.5. Near-field communication using inductive coupling.	23
Figure 3.6. Far-field communication using radiative coupling.....	24
Figure 3.7. An example of some commercial UHF passive tags.....	24
Figure 3.8. Typical RFID signal strength over distance [47]	26
Figure 3.9. Power intensity of a RFID reader in an empty room[48].....	27
Figure 4.1. Example of a costmap.	30
Figure 4.2. Pose probability density functions of LIDAR, Kinect, and the odometry.	33
Figure 4.3. Outlines multiple- layer mapping.....	34
Figure 4.4. The output of the global path planner at various stages.	38
Figure 4.5. Sensor limitation.....	41
Figure 4.6. The local planner will generate multiple routes, and RFID inventory quality is maximized by the one closest to the merchandise.	42
Figure 5.1 RFID model.....	47
Figure 5.2. Procedures of fixed power tag localization algorithm.....	50
Figure 5.3. The RFID model for Power lever 170, 140,110 and 80.	54

Figure 5.4. Variable power RFID model.	55
Figure 6.1(a) the top view blueprint map, and (b) photo of the store.	58
Figure 6.2(a) Experimental RFID tag Avery 237, and (b)The RFID tag attached to a merchandise item	59
Figure 6.3. The main parts of our experimental robotic system.	59
Figure 6.4. Robot components	60
Figure 6.5(a) photo of shoe rack, (b) depth image of shoe rack	62
Figure 6.6(a) SLAM with LIDAR scans only, (b) SLAM with Kinect scans only	63
Figure 6.7. Fused map by multiple-layer mapping algorithm with both LIDAR and Kinect scans.	64
Figure 6.8. Inflated costmap of the experimental mockup store.....	65
Figure 6.9(a) the global path generated with $l=0.45$, and (b) the global path generated with $l=0.8$	66
Figure 6.10(a) Robot inventory navigation result of the closer path, (b) Robot inventory navigation result of the farther path	67
Figure 6.11(a) Zebra RFD8500 handheld, (b) waving handheld in front of merchandise to do manual inventory.	68
Figure 6.12. Inventory accuracy chart	70
Figure 6.13(a) Illustrates the robot cannot detect the high level RFID tags of the jean shelf, (b) In the photo the robot navigates and collects RFID tags response in front of the jean shelf.	71
Figure 6.14. Jean shelf level inventory accuracy chart.....	73
Figure 6.15. The blue points represent the measurement points of variable power tag localization	74
Figure 6.16. Tag localization rate by different algorithms	75
Figure 6.17. Tag localization average error and standard deviation by different algorithms	76
Figure 6.18. Examples of localized items in our mock store.....	78

List of Tables

Table 4.1. The outline of the global path planner algorithm	36
Table 4.2. Outline of local path planner	39
Table 5.1. Pseudo-code of one step of Bayesian updating	44
Table 5.2. The tag location estimating of fixed power tag localization algorithm	48
Table 6.1. Inventory accuracy.....	69
Table 6.2. the jean shelf level inventory accuracy.....	72
Table 6.3. Tag localization results	75

List of Abbreviations

ROS	Robot Operating System
SLAM	Simultaneous Location and Mapping
LIDAR	Light Detection and Ranging
RFID	radio-frequency identification
EPC	Electronic Product Code
UPC	Universal Product Code
LF	low-frequency
HF	high-frequency
UHF	ultra-high frequency
NF	Near Field
IFF	identify friend or foe
AP	Access Point
ISO	International Standards Organization
IEC	International Electrotechnical Commission
ANA	Article Number Association
EAN	European Article Numbering
UCC	Uniform Code Council

Chapter 1 Introduction

Since the emergence of low-cost and reliable RFID tags, the retail world is undergoing a rapid transition from bar code to RFID [1]. A number of consequent benefits are available to the retailer and the customer, but these are not fully realized unless frequent, accurate inventory can be performed. Conventional RFID-based inventory counting, which is called “cycle counting,” requires people to walk around the store or warehouse using hand-held scanners to collect the RFID tags responses. The process is not particularly well-defined – each individual tends to develop his/her own approach that seems to yield the best results in the shortest amount of time. It generally involves walking around racks and shelves while waving the scanner. The scanner provides audible feedback (“popcorn”) that enables the user to determine when no new tags have been detected in a particular area. This process is time-consuming and labor-intensive. It also provides a relatively static picture of the inventory, and does not provide a convenient way to determine the location of items in the store.

Meanwhile, in the last decade considerable progress has been made in the development of autonomous robots. The use of robots is increasing in environments relevant to the everyday life of the average person, such as tour guide robots [2] or hospital service robots [3]. Furthermore, since the introduction of the first commercially available vacuum cleaner robot, Electrolux Trilobrite [4], more and more families have begun to accept and use vacuum cleaner robots. The Roomba vacuum of iRobot is the most prevalent one, having entered the market with overwhelming success. The success of these robotic vacuum cleaners is just the tip of the

iceberg for what autonomous mobile robots are capable of doing, and we believe more and more autonomous robots will service us in our daily life.

Autonomous robot research has focused on applications that are too monotonous or dangerous for humans to perform. An autonomous lawn mower was built by Auburn University [5]. It is capable of cutting around fences, and during operation it can autonomously avoid dynamic obstacles such as a person walking in front of it. Underwater robots are designed to explore the deep oceans, as well as rivers, lakes, and costal waters [6]. As we mentioned previously, performing retail inventory is a typical boring, time-consuming and labor-intensive task for a human being. Hence, deploying an autonomous robot to do inventory in a retail environment has captured the interest of retailers. Retailers are getting serious about deploying robots on the sales floor and in the stockroom for various aspects of inventory management. There are several pioneering robotic applications reported for similar purposes.

Ehrenberg describes a system called LiBot, which automates the shelf-reading process and finds misplaced books autonomously in a library [7]. The system determines the order of the books on the shelves, and localizes each book. The paper addresses issues such as metal shelves and thin books with RFID tags in close proximity, which affect location accuracy. To deal with the limited location accuracy of the HF RFID system and the resulting uncertainty about the true position of the RFID tags, the authors used a probabilistic approach based on recursive Bayesian updating. Various limitations are described in the paper, primarily due to idealizing assumptions about the placement of tags, and the ability of the robot to maneuver precisely.

Schairer presents a system for machine-aided inventory, along with results from testing in a simulated supermarket environment [8]. The robot, equipped with an RFID reader, traverses the supermarket environment while constantly detecting products within its range. The data are

transmitted using WLAN to a central computer which holds a model of the current state of the system. These data are augmented by additional product-specific information provided by an Enterprise Resource Planning (ERP) system. The detected objects as well as additional product data are visualized using Augmented Reality (AR) techniques. The robot localizes itself using the existing infrastructure of different ambient sensors (RFID, Bluetooth, WLAN).

Melià -Seguí describes an actual installation in an apparel retail store in Barcelona, and presents some of the initial conclusions after several months of operation with real customers. In this case the RFID-based inventory is performed [9].

1.1 Goals

The work presented in this dissertation aims to develop an autonomous robot system to process RFID based inventory and localize the RFID tagged merchandise. In such a system, an autonomous robot equipped with RFID scanner and several antennae is deployed to cover all the merchandise-containing space of a retail store. The robot will autonomously navigate in the retail environment to scan RFID tags and localize each tag, therefore localizing the related item. A key feature of the navigation algorithm is the ability to intelligently plan paths that bring the robot near areas to be scanned, as opposed to wasting time in open space. In a sense this is the complementary behavior to the well-studied, classical coverage problem in mobile robotics. The latter would be used, for example, in a robotic vacuum cleaner, or an autonomous agricultural tractor. In the present instance, we want to guide the robot near shelves and fixtures that contain merchandise, while minimizing the time and energy spent traversing open areas. This technology offers the possibility to perform inventory much more frequently than would be cost-effective for manual operations.

In this dissertation, we also propose two RFID tag localization algorithms for our robot; they are referred to as fixed power localization and variable power localization. Both algorithms are probabilistic in nature. The goal is to accurately report the location of multiple passive RFID tags in a target place. Passive RFID is not a precise location technology; we will explain more about this in Chapter 3. The fixed power tag localization algorithm will collect RFID tag responses at a fix power of the RFID reader. After collecting enough responses for an RFID tag in different positions, the fixed power tag localization algorithm could estimate the location for this tag. The greatest complicating factor to affect location accuracy is radio wave reflection (multipath fading). One approach to solve this problem is varying the power of the reader through the antenna to reduce multipath. When the successful tag responses are measured at varying power levels from multiple locations in the test area, the variable power tag localization algorithm can be used to precisely locate the RFID tag. For example, when a tag can be read at a high power level, but cannot be read at a low power level, we can determine relative range of the tag to the robot. The goal of the variable power algorithm is that we can determine the location of passive RFID tag in a retail environment with accuracy of 0.5 meter.

1.2 Motivation

The first challenge of an autonomous robot operating in a retail environment is how to handle a highly unstructured and dynamic environment. To perform the task, sensors must be used to perceive the environment surrounding the robot. Sonar, laser range-finders and depth image cameras have been the most prominent technology utilized in mobile robot community. Many indoor robot applications deploy a 2D map to represent the surrounding environment that is built by the mounted sensors. Compared to a 3D map, the 2D map consumes very little computing resources which means the autonomous navigation could be deployed with a cost

effective hardware platform. The drawback of a 2D map is that it only provides a single vertical slice representative of the environment. This drawback will limit the utility of an autonomous robot operating in a retail environment which is highly geometrically intricate.

To address this problem, we introduce a multilevel map which represents the environment at multiple heights. Such a map is a practical departure from the ideal of a complete 3-D map, which is intractable computationally. It provides rich vertical information for the environment while keeping the advantage of limited computing resource consumption. In order to deploy a mobile robot to perform retail inventory autonomously, a key feature of the navigation algorithm is the ability to intelligently plan paths that bring the robot near areas to be scanned, as opposed to wasting time in open space. We propose an inventory based path planner algorithm that can efficiently guide the robot to cover whole target spaces to collect RFID tag responses.

On the other hand, the localization of passive RFID tags, which are dominant in retail, is another challenge for the RFID community. The intricate environment will prevent any existing wireless localization algorithm to be deployed for localizing passive RFID tags in the retail environment. We propose two RFID tag localization algorithms, fixed power and variable power tag localization algorithm, to handle this task. The fixed power tag localization algorithm is more efficient than the variable power tag localization algorithm, however, the localization accuracy is lower than the variable power one.

The remainder of this dissertation is organized in the following manner: Chapter 2 provides an overview of the field of autonomous mobile robots and wireless localization. Chapter 3 introduces the background of RFID technology. Chapter 4 details the whole navigational algorithm for an autonomous robot to do inventory in a retail environment. Chapter

5 presents the RFID tag localization algorithm. Chapter 6 introduces experimental setup and results, followed by the conclusion and suggestions for future work in Chapter 7.

Chapter 2 Literature Survey

Last decade witnessed the bloom of robotics research, and various robots appeared in our daily life. The field of robotics has become a vast and multidisciplinary thrust at research institutions around the world. The main research topics of robotics include: socially assistive robots [10], personal home automation robots [11], industrial manufacturing robots [12], search and rescue robots [13], and many more. During all these researches the navigation strategy is the fundamental problem for all mobile robotics. In this chapter, we introduce some key research concepts of robotics with an emphasis on the mobile robot navigation issue.

Also in the last decade there has been much research focus on the RFID tag localization issue, especially since the RFID technology has been widely accepted by retail and industrial manufacturers since the beginning of the twenty first century. The RFID tag localization can be categorized as a famous research topic related to wireless localization [14]. This chapter also introduces some important research concepts around indoor wireless localization and mostly focuses on the RFID tag localization issue.

2.1 Motion Planning of Autonomous Mobile Robots

The original goal of motion planning algorithms [15] was to solve the start-goal problem. These algorithms find out solutions for determining a path or trajectory between the start and goal point. Many papers talk about this topic, and they can be roughly divided into two categories: global and local motion planning. The global motion planning assumes that a complete environmental model is available, and it generates a path based on the knowledge of

this environmental model. While the local motion planning model only requires modeling a small part of the environment to generate a path.

Moravec demonstrated a probabilistic method to update global world models based on sensory input of a robot, and then based on these world models to generate a route for the robot [16]. In his work, a Uranus mobile robot deploys a Bayesian statistical foundation to maintain a probabilistic, geometric map of the robot's surroundings as it moves. The robot can deal correctly with uncertainties utilizing a continuously updating map. The updating map helps the robot to choose clear paths, identify locations (by correlating maps), and to identify well-known and insufficiently sensed terrain.

Hu and Brady describe a global path planner algorithm that enables a robot to respond to unexpected static obstacles [17]. In their work, they present a sensor-based method for real-time obstacle avoidance while a robot is navigating. It aims to achieve both collision free motion and minimum cost. The approach utilizes Bayesian decision theory to determine an optimal response to obstacles based on inaccurate sensor data. The optimal navigating rule minimizes the Bayes risk by trading between an alternative route or a sidestep maneuver for the current path.

Ó'Dúnlaing, Colm, and Yap present a new method for path planning that uses a map of the environment [18]. Their map is a generalized Voronoi diagram. Their method reduces the search for a collision-free motion by searching for a connected path along the edges of such a diagram. The approach could plan an obstacle-avoiding motion of a single circular disc amid polygonal obstacles.

A vector field histogram approach is proposed by Borenstein and Koren [19]. This method is based on the virtual force field histogram [20] that they previously developed. In their work, an occupancy grid map is deployed for modeling the environment. This map is generated

and updated continuously by the robot's ultrasonic sensors. The approach checks the free spaces by transforming the occupancy map into a histogram, then the path planner will generate the motion and velocity for the robot.

The Dynamic Window Approach (DWA) [21], which is the most prevalent local path planner, was proposed by Fox, Burgard and Thrun. The DWA provides a dynamic windows method to reactive collision avoidance for a synchro-drives mobile robot. The DWA directly searches the controlling command in a velocities space, it also take the dynamics limits of robot into account. The DWA only detects the potential collision in a very short time period, and only considers admissible velocities to generate a trajectory on which the robot is able to stop safely.

Besides the conventional start-goal path planning algorithms, there is a subdivision of robot motion called coverage path planning algorithms [22]. These algorithms address the applications for floor cleaning [23], lawn mowing [24], mine hunting [25], etc... This kind of path planning algorithm controls robots to sweep out the whole target spaces physically or by its sensors.

A floor coverage approach is described by Hofner and Schmidt [26]. This approach takes the non-holonomic constraints into account. This approach uses a set of patterns to cover a bounded region, which is free of obstacles. The robot's path is represented by a concatenation of motion patterns. These patterns accommodate the non-holonomic constraints of the robot.

Elfes presents an approximate cellular decomposition method for the robot coverage problem [27]. The approach is developed for an autonomous robot navigating in an unknown and unstructured environment. It uses a fine grid based map to represent the environment, the cells are the same size and shape, and it assume that if the robot enters a cells it has covered this cell.

Lumelsky demonstrates a coverage algorithm that is based on a partial discretization of space [28]. The space is represented by cells with fixed width while the top and bottom can be any shape. The planar terrain-covering algorithm can navigate the robot in a connected environment to cover all cells of the map, and the robot could start at an arbitrary point.

2.2 Wireless and RFID Localization Algorithms

An astonishing growth of wireless systems has been witnessed in the last decades. As wireless system bloom, the demand for accurate positioning in wireless networks has been growing. The wireless localization can be roughly divided into two categories, indoor and outdoor wireless localization [29,30]. The primary progress in RFID systems has been made during the last ten years. The RFID localization researches can be categorized as wireless localization, because they share many common characteristics. It is very difficult to model radio wave propagation in the indoor environment due to the severe multipath and low chances of line-of-sight (LOS) path. Therefore, there is no good model for indoor radio multipath so far [30].

Time of arrival (TOA) is one of the most famous algorithms in wireless location. It measures the radio signal propagation time to evaluate the distance. TOA measurements must be made with respect to signals from at least 3 reference transmitters. The basic idea is shown in the following figure [31].

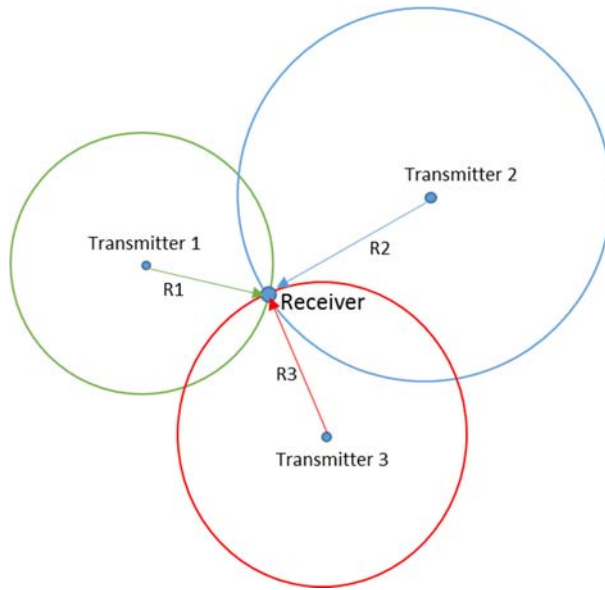


Figure 2.1. Positioning based on TOA

There are two preconditions for TOA working, firstly, all transmitters and receivers have to be precisely synchronized. Secondly, the signals must provide timestamp. To overcome the drawback of precisely synchronizing between transmitters and receivers, time difference of arrival (TDOA) is introduced [32]. TDOA determine the relative position by examining the difference of receiving time of signals from multiple transmitters, instead of absolute time of TOA. TDOA and TOA algorithms do not work with UHF passive RFID, because of the narrow bandwidth of UHF passive RFID, which is 902 – 928 MHz in United States. RFID readers and tags can not operate in short pulse mode required for TDOA and TOA distance determination [33].

RSS-Based (or called Signal Attenuation-Based) method estimates the distance of a receiver to a transmitter by using the attenuation of emitted signal strength. However server multipath fading of indoor environment greatly reduce the localization accuracy and the parameters employed in RSS-Based models are site-specific. There are some methods that can be used to improve accuracy, such as utilizing the premeasured RSS contours centered at the

receiver [34] or multiple measurements at several base stations. Teuber, Eissfeller and Pany introduce a fuzzy logic algorithm that is able to significantly improve the location accuracy using RSS measurement [35].

Received Signal Phase Method or POA [30] utilizes the received radio wave phase or phase difference to estimate the distance between a receiver and transmitters. It needs all transmitters to emit sinusoidal signals at the same frequency and same phase. As the receiver is located at an arbitrary position, the distances between it and each transmitter are different. The different distances will cause different receive phases, and we can use these phases to estimate the distances. With distances to at least three transmitters, we can use similar method or TOA to evaluate the position of the receiver.

The angulation Techniques or AOA estimation evaluates [36,37] the location of target by the intersection of several pairs of angle direction lines which are the radius from the transmitters to receiver. The basic idea of AOA is given in the following figure.

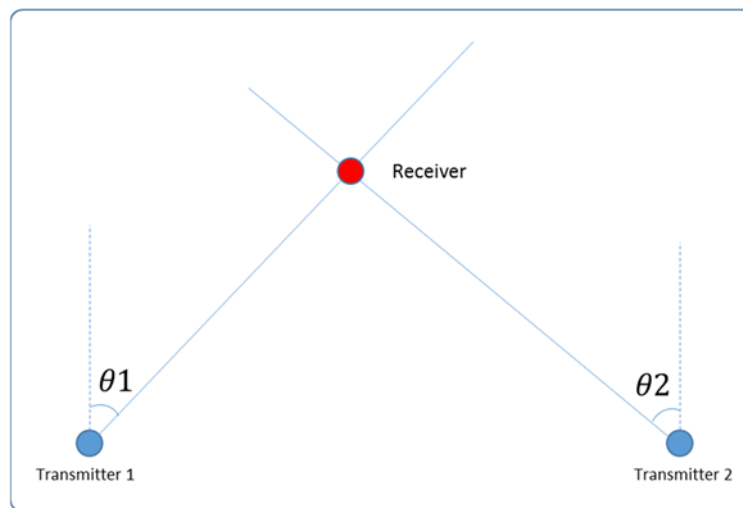


Figure 2.2. Positioning based on AOA

Xuyu Wang demonstrates an AOA based method, which uses cooperative APs with antenna arrays for accurate indoor localization [38]. This method first estimates the arriving

angles for all multipath components using the MUSIC (Multiple Signal Classification) algorithm. Then it exploits the geometric relationship among the angles to identify LOS (Line of Sight) angles. The user location can be computed with the LOS angles and the known distance between the two APs.

In addition to the traditional approaches, machine learning based methods became a hot topic in wireless localization. Multiple researchers [39, 40, 41] proposed a deep learning based indoor fingerprinting system. The system consists of an off-line training phase and an on-line localization phase. In the off-line training phase, the deep learning is utilized to train all the weights of a deep network as fingerprints. Then in the on-line localization phase, a probabilistic method based on the radial basis function is used to obtain the estimated location.

Chapter 3 RFID Technology Overview

The retail world is undergoing a rapid transition from bar code to radio frequency identification (RFID) [1]. The motivation for the transition is to overcome the deficiencies of existing bar code technology. The sensing device requires a clear line of sight to bar codes. This deficiency greatly reduces the efficiency of inventory counts, especially in giant warehouse and sale floors. The bar codes can be distorted or defaced by dirt, paint, ink and other opaque materials. Mechanical damage to the bar codes degrades their readability, as well. Data stored by the bar code on a surface is not readily modified or extended, save perhaps by wholesale replacement. Finally, most of the bar codes only provide product level identification. Items with the same product category share the same Universal Product Code (UPC). It cannot provide item level identification which is a fundamental requirement for product tracking and quality control.

To remedy these deficiencies of bar code technology, the retail sector has turned to an alternative technique, radio-frequency identification (RFID). While RFID has been around for decades, the dramatically decreased cost and increased capabilities have motivated the retail world to widely deploy this technology. RFID offers many benefits for supply chain management, inventory control, and many other applications. It provides item level Electronic Product Code (EPC) that is unique for each item. RFID, especial the UHF passive type that relies on the radio wave to communicate between reader and tags, provides greater convenience for inventory, and greatly reduces the manual labor of inventory management.

In this chapter, we introduce the history of RFID. Then we categorize RFID tags. Thirdly, we examine the technical basis for passive RFID. Lastly, we introduce some characters of passive RFID tags and illustrate how they affect the robot inventory and RFID tag localization.

3.1 History of RFID

The basic idea of RFID technology, communication using reflected radio energy, can be traced back to the origin of radar technology. During World War II, the radar was deployed by allies and the enemies to detect planes while they were still miles away. Unfortunately, one problem with radar technology was that there was no way to identify the type of plane. The Germans discovered that if they rolled the airplane when they returned to base, it would change the reflected radar waves, and they used this method to inform the radar operators that these were German planes [42]. This phenomenon, essentially, worked as the first passive RFID system. On the other hand, the British developed the first active system to identify friend or foe (IFF). A transmitter was deployed on each British plane, when it received the signal from the radar, it replied with a signal that allowed for the identification of the plane [42]. The modern RFID system works the same as this basic concept, when a signal is sent to a RFID tag, it wakes up and either reflects back a signal (passive RFID tag) or sends a signal (active RFID tag).

In 1948 Stockman published a paper to introduce a work related to RFID that involved continuous time modulation of reflected signals. He designed a device that could modulate human voice on reflected light signal [43]. RF communication systems greatly improved throughout the 1950s to 1960s, resulting in the deployment of the first anti-theft system based on radio waves. This system used radio waves to determine whether an item had been paid for or not. It attached an electronic article surveillance tag to an item; these kind of tags are still used today. The tag is a 1-bit tag, which is either on or off. If the item is paid for, the bit of tag will be

turned off, otherwise the bit will remain on. When the item goes out of a store, readers at the doors will detect the tag and sound an alarm if the tag bit is still on [44].

During the 1960s to 1970s, there was more research in the RFID area. The most significant break-through in this period was made by Robert Richardson. He developed a device that could couple and rectify the energy from an electromagnetic (EM) field and transmit signals at a harmonic of the received frequency, which is the basic concept of modern passive RFID tag to extract power and the way to send signals [42]. Later in the same decade, Vinding developed a system that was based on inductive coupling technology, which is the ancestor of today's Near Field (NF) RFID system [42].

In the 1970s, Los Alamos National Laboratory made a great contribution to RFID system. In 1975, Koelle, Depp and Freyman of Los Alamos introduced a novel concept that the transponder antenna load modulation can work as a simple and effective way for backscatter modulation [42]. They developed a system that placed a transponder in a truck and readers at the gates of secure facilities. They used this system for tracking purposes, the gate antennae would wake up the transponder in the truck, and then the transponder would respond with an ID and other data for tracking, like the driver's ID [44]. This system was commercialized in the 1980s, and it became widely deployed on roads, bridges and tunnels. Los Alamos also developed a passive RFID tag for tracking cows. They developed a passive RFID system that used Ultra high frequency (UHF) radio waves, in which the tag extracts energy from the radio wave and simply reflects a modulated signal back to the reader for identifying the cow. After that, companies developed a low-frequency system, which was 125 kHz, and developed smaller tags that could be encapsulated in glass. These encapsulated tags could be injected under the cows' skin. The

similar system is still used in the livestock industry today [44]. The following figure illustrates the idea of tagging a RFID tag to a sheep's ear.



Figure 3.1. A sheep with a RFID tag, the tag is marked by a red rectangle

After the success of the 125 kHz RFID systems, many companies moved up to higher frequency, 13.56MHz, which could offer greater range and faster data transfer rate. Today, 13.56 MHz RFID systems are widely used for access control, payment systems (Mobile Speed pass) and contactless smart cards. The automobile industry also introduced the 13.56 MHz RFID system as an anti-theft device in cars. A reader in the steering column of a car reads the passive RFID tag in the key. The reader will check the ID number feedback from the key tag, and compare it with the programmed ID, if they do not match the car will not start [44].

In the early 1990s, an ultra-high frequency (UHF) RFID system was developed by IBM, it offered up to 20 feet read range and higher data transfer rate. Later, this technology was sold to Intermec, a bar code systems provider. Then, Intermec's RFID systems were deployed in warehouse tracking, farming, etc. But the high cost of RFID systems prevented the widespread deployment of RFID technology. The RFID technology was expensive at the time due to the low volume of sales and the lack of open, international standards [44].

The 1990s to 2000s witnessed a sharp increase in the RFID international standards activity. Many standardization activities were conducted in the 1990s, most of these were conducted by the International Standards Organization (ISO) and International Electrotechnical Commission (IEC). Initial standardization interests were in animal tracking (ISO-11784 and ISO-11785) and contactless proximity card (ISO-14443) applications. In the 1990s, RFID technology was accepted by supply chain management, and this motivated more standardization activities. In 1996, the standardization of RFID was accepted by the Article Number Association (ANA) and European Article Numbering (EAN) groups. In 1999, a UHF frequency band for RFID was adopted by EAN international and the Uniform Code Council (UCC) of the United States. They also established an Auto-ID Center at the Massachusetts Institute of Technology. This organization developed a global RFID standard for product labeling called the Electronic Product Code (EPC), this standard enabled the capability of item level identification. During this period, the biggest improvements resulted from the way people used the RFID. Previously, tags were used as a mobile database that stored information about the product. Sarma and Brock [44] only stored a serial number on the RFID tag, and the information of related product associated with the serial number was stored in a database that would be retrieved through the Internet. They turned RFID into a networking technology by linking products to the Internet through the tags.

With the growth of silicon technology, RFID tags became cheaper and more reliable. In the first decade of the 21st century, the world moved toward the technology's widespread and large-scale adoption. Some of the biggest retailers in the world—Albertsons, Metro, Target, Tesco, Wal-Mart—and the U.S. Department of Defense have said they plan to use EPC

technology to track goods in their supply chain. And many other industries are also moving to adopt the RFID technology.

3.2 Architecture of a RFID System and Type of RFID

A typical RFID system consists of an interrogator, more often called a reader, transponders, also called tags, and the antennae that are connected to the reader for sending and receiving signals. Figure 3.2 shows a typical RFID system. The antennae may be integrated with the reader, like a hand-held system or separate from the reader and connected by a cable. The tag is integrated with the antenna and at least one silicon chip, which responds to the reader query and stores data to built-in memory. In most cases, the reader will be connected to a computer to provide interface with other systems or human beings.

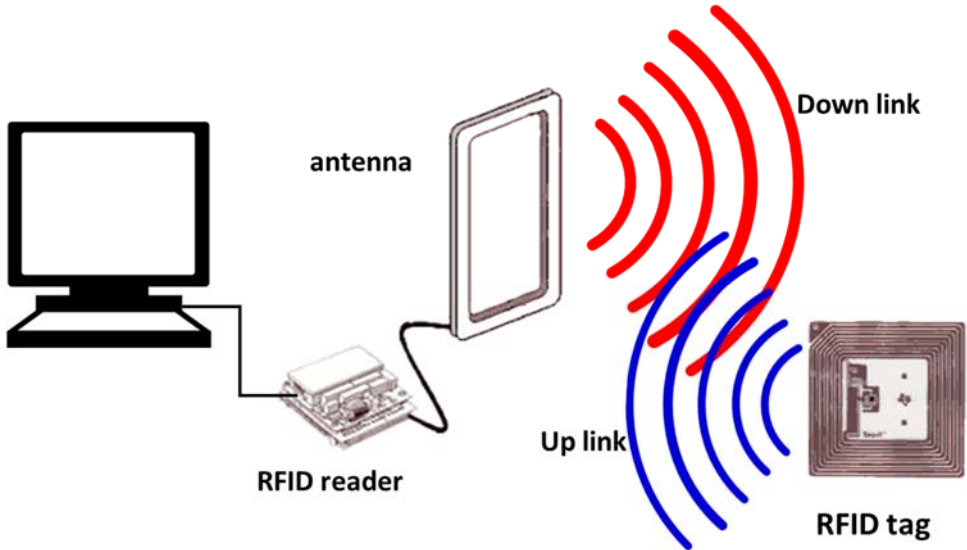


Figure 3.2. A typical RFID system

A RFID system can be classified by the frequency of the radio waves they use and by power source of the tag. RFID systems can use frequencies from around 100 kHz to over 5 GHz, the RFID frequency bands are shown in Figure 3.3. The frequency of most RFID systems are concentrated in very narrow bands that are available by regulators for unlicensed industrial

implementation. The most commonly use frequency bands are 125/134 kHz, 13.56 MHz, 860-960 MHz and 2.4 GHz. The RFID systems that rely on 125/134 kHz radio wave are called low-frequency (LF) systems. While systems that work on 13.56 MHz are referred to as high-frequency (HF) systems. Systems that operate in 860-960 MHz are called ultra-high-frequency (UHF) systems, and the systems in 2.4 GHz are referred to as microwave systems. Nowadays, most retail deployments of UHF RFID system are in their warehouses and sales floors environments.

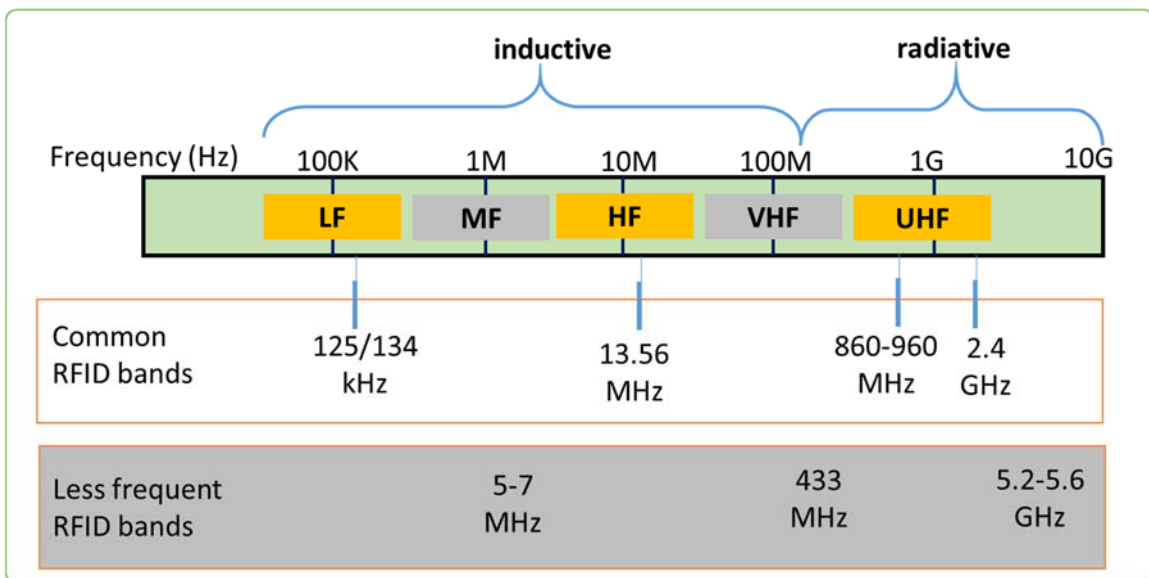


Figure 3.3. Typical frequency spectrum available for RFID

From the power source of the tag, the system can be classified into three categories as follows:

1. Passive RFID system
2. Semi-passive RFID system
3. Active RFID system

In a passive RFID system, the tag has no 'on-board' power source. The tag has to extract power from the signals sent by the reader. This kind of tag is usually small and low cost, so it is the most

widely used in retail. In an active RFID system, the tag has an on-board battery. The ‘on-board’ battery provides power for transmitting the data from the tag to the reader. The battery also provides power for other electronic components of the tag. A semi-passive tag is actually a passive one in nature, but it contains a battery to supply power to the built-in chip and auxiliary components like sensors, user interface etc. Figure 3.4 shows the difference among passive, semi-passive and active RFID system.

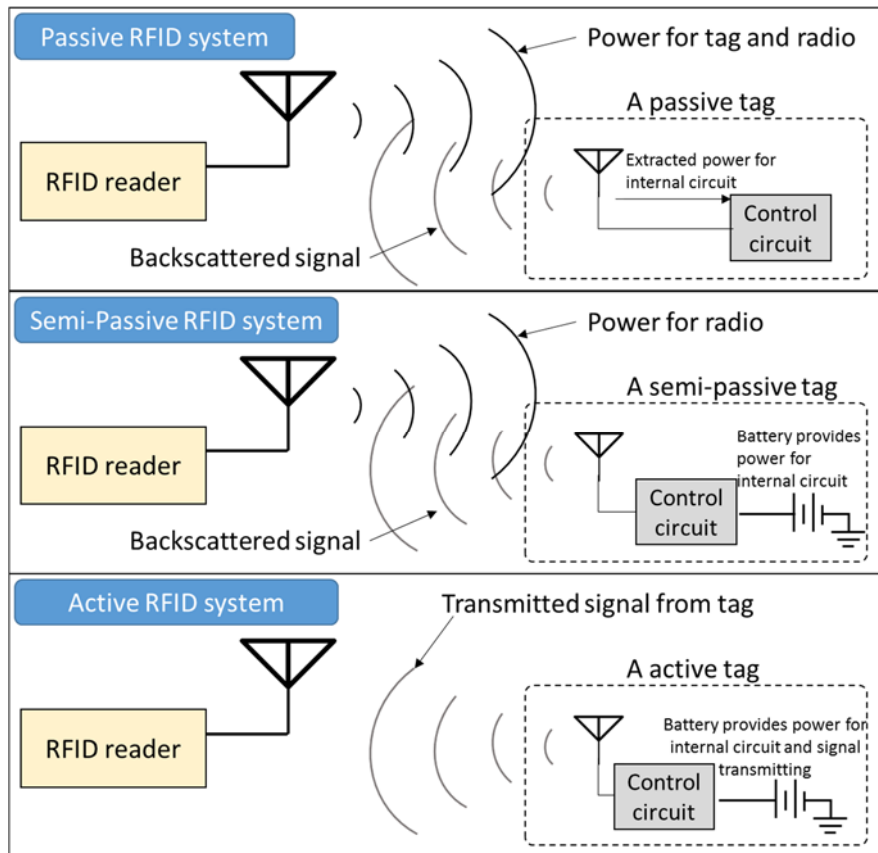


Figure 3.4. Passive, semi-passive and active RFID system.

In our work, we will focus on the passive RFID system, because UHF passive RFID systems are most widely deployed in the retail sector. This is mostly because of the low cost and reliability of UHF passive RFID system. Later in this chapter we will introduce more technical details about the passive RFID system.

3.3 Technical Basics for Passive RFID

Without a built-in battery the passive RFID tag extracts power from the electromagnetic field that is created by the reader. The extracted energy is used to power up internal circuits. There are two different power extracting techniques or coupling techniques, inductively coupling and radiative coupling, used by passive tags.

Inductively coupling is usually used in the near-field (NF) region. Most of near-field (NF) tags are inductive coupled to a reader antenna coil by the magnetic field. This mechanism is based upon Faraday's principle of magnetic induction. When a current flowing through the coil of the reader antenna, it will produce a magnetic field around it. Then, this magnetic field causes a nearby tag's antenna coil to generate a current, this phenomenon is called magnetic induction [45]. The communication between the reader and a tag relies on load modulation [46]. Based on the mutual inductance between the reader antenna coil and the tag coil, any current variation from the tag coil causes a current variation in the reader's coil. This current variation can be detected by the reader, hence a tag could change its current to communicate with the reader. The current in the tag coil is proportional to the load applied to the coil, so we can change the load to change the current, and this technology is called load modulation. The basic idea of near field communication by inductive coupling is shown in Figure. 3.5.

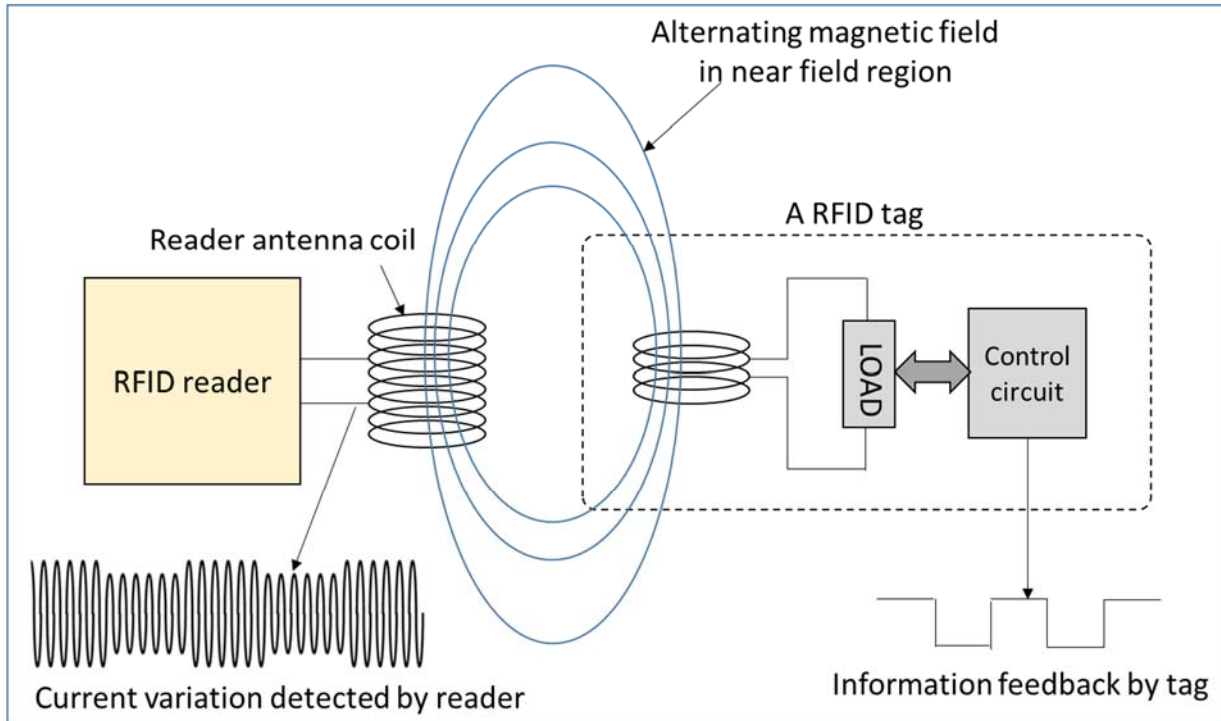


Figure 3.5. Near-field communication using inductive coupling.

Generally speaking, the inductive coupling RFID works on low frequency, most common frequencies are 128 kHz and 13.56 MHz. A drawback of inductive RFID system is that it only can detect tags within a very short distance, because the power of the magnetic field drops as $\frac{1}{r^6}$, where r is the distance between a tag and the reader antenna coil.

Radiative coupling is applied in the far-field. A tag extracts power from the radio wave that is sent by the reader antenna. This extracted power will be used to power up internal control circuits. A method called backscattering is used to communicate between the tag and the reader. Part of the radio wave will be reflected back to the reader antenna, due to the impedance mismatch between the tag antenna and the load. So changing the load of the tag could vary the strength of the reflected wave. The tag changes in load could send data through the varying reflected wave, which can be detected by the reader. This method is called backscattering. The basic idea of radiative coupling and backscattering are illustrated in Figure. 3.6.

3.4 Characteristics of UHF Passive RFID

In this section we will introduce two characteristics of UHF passive RFID; detectable range and multipath fading. When we design the robot to do autonomous inventory and localize RFID tags, we must take these two characteristics into account.

3.4.1 Detectable Range of UHF Passive RFID

The first characteristic we are interested in is the detectable range of a UHF passive RFID reader. A passive RFID tag without any built-in power supply, absorbs power from radio wave that is sent by the reader, hence the power intensity of radio wave determines the detectable range of a UHF passive RFID system. In an unobstructed free space, the radio wave strength falls off as the square of the distance traveled, which can be determined by Friis transmission equation

$$P_r = P_t \frac{G_t G_r \lambda^2}{16L\pi^2 d^2}$$

where P_r is the power received by a RFID tag antenna, P_t is the power input to the RFID reader antenna, G_t is reader antenna gain, G_r is RFID tag antenna gain, L is system loss factor, λ is the wavelength, and d is the distance between the reader antenna and a RFID tag. So, the reader can only read the tag within a specific distance, and as the distance between the reader and a tag increasing the power intensity decreases rapidly. A typical RFID antenna power attenuation is illustrated in Figure 3.8.

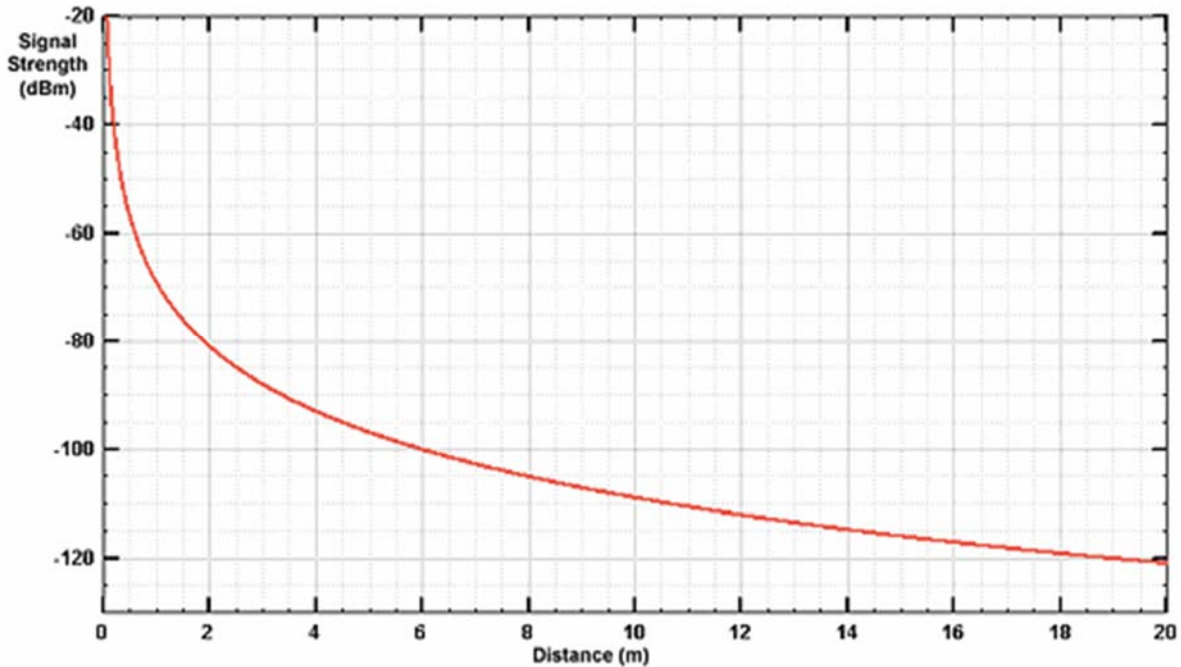


Figure 3.8. Typical RFID signal strength over distance [47]

From the above figure, we can see that as the distance between RFID tag and RFID antenna increases, the received power by tag decreases tremendously, consequently the probability of detecting that tag decreases tremendously. So, when our robot does inventory, it needs to move as close as it can to read more tags.

3.4.2 Multipath Fading of UHF Passive RFID system

As we introduced above, in free space, the power intensity of UHF passive RFID radio wave falls off as the square of the distance traveled. But in a real retail environment, we cannot predict the power intensity by the Friis transmission equation at all. The radio wave of UHF passive RFID system reflects from obstacles. The reflected waves propagate and interfere with the waves sent by the reader antenna; this phenomenon is called multipath fading. In a retail environment, many items such as shelves, racks, and even ornaments will reflect the radio waves, and create a very complex radio power intensity field, which almost cannot be predicted. We take a very simple environment as an example to demonstrate how the multipath fading will

affect the radio power intensity field. A RFID reader is placed within a simple rectangular room, the wall and floor will reflect radio wave, and there are no other obstacles. The power density field is shown in the following figure [48]

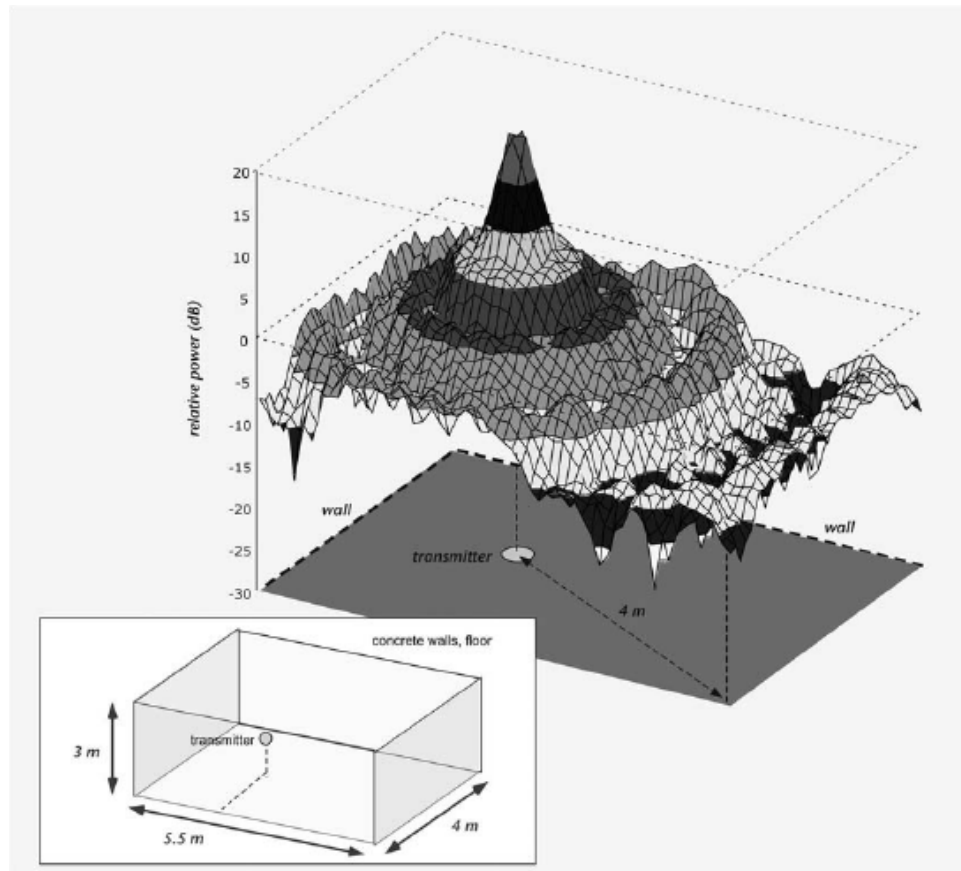


Figure 3.9. Power intensity of a RFID reader in an empty room [48]

From the above figure, we can tell the power density decreases monotonically near the reader, however after distances greater than 1 meter the power density becomes very complex. In such a room, by moving a tag away from the reader antenna, we could find that a tag may disappear and then reappear. And in a real retail environment, the power intensity field is even more complex. The multipath fading is a big challenge to our RFID tag localization, because the complex power field creates a complex RFID detectable model, which is a fundamental component for RFID tag localization.

Chapter 4 Robot Navigation for Retail Environment

Efficiently and safely navigating throughout the retail environment is an essential task for a robot during inventory and RFID tags localization. An autonomous robot needs to cover all of the potential space in this environment to collect responses from the RFID tags. In this chapter we will detail an approach for the robot to navigate and cover the whole target space. Robot navigation is a vast field, and it can be divided into three subcategories.

- How to represent the environment:

A robot detects its environment by its sensors. An autonomous robot must answer a fundamental question: given the number of noisy sensory observations, what does the environment actually look like? A map built by sensor observations is a common way to represent the environment.

- How to localize the robot:

Throughout the last decade, sensor-based localization has been recognized as a key problem in mobile robotics. It answers a critical question for the robot: given some sensor observations and a map, can the robot localize against the given map? If the robot can correctly localize itself, it can properly navigate itself in a given map.

- Robot path planning:

One of the ultimate goals of mobile robotics research is to build robots that can safely and efficiently navigate indoor environments. Robots must localize themselves by their sensor

observations and make a path plan that can guide itself to its goals. A robot must react to unforeseen circumstances, just like a human.

There is one other requirement for a robot that works in a retail environment. That is how efficiently it covers the whole target space. Merchandise items that are tagged with RFID tags usually are distributed throughout the whole retail environment. The robot must cover all the target spaces to collect the RFID tag response by moving closer to them. In this chapter we will present the details of our robot navigation algorithm, which could answer all the above questions.

4.1 Mapping

In this dissertation we employ the well-known costmap strategy, which is kind of like an occupancy grid map, to represent the environment. Occupancy grid maps are spatial representations of the robot's environments. They represent environments by fine-grained, metric grids of variables that reflect the occupancy of the environment. In a costmap, each cell can have one of 255 different cost values, to represent the free, occupied or unknown statuses. Figure 4.1 is an example of a costmap. The black cells represent obstacles, and the colored cells represent obstacles inflated by the inscribed radius of the robot. The red circle represents the footprint of the robot, while the white represents the free space, and the gray represents the unknown space. To avoid collision, the footprint of the robot should never intersect a black cell and the center point of the robot should never cross a colored cell.

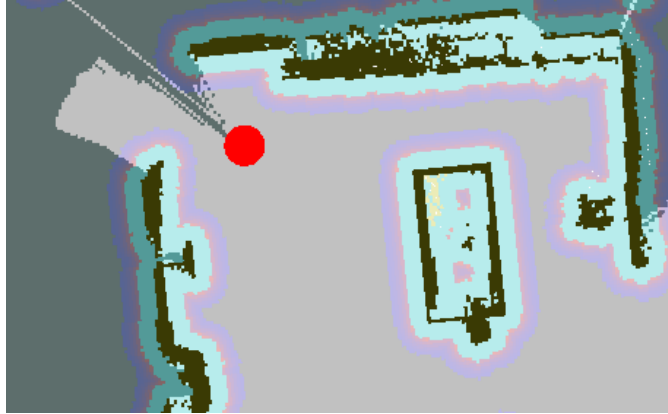


Figure 4.1. Example of a costmap.

Our map is generated from the observations of the robot's sensors, while manually controlling the robot to navigate throughout the room. One of the most important skills enabling robots to navigate efficiently in dynamic and obstacle-rich environments is the ability to generate maps at multiple heights; i.e., multilevel maps. Such a map is a practical departure from the ideal of a complete 3-D map, which is intractable computationally. The process of building a map dynamically in a partially unknown environment, and locating oneself within it, is called simultaneous location and mapping (SLAM) [49]. This technique is well-established in the robotics community; in fact, it is a fundamental component built into Robot Operating System (ROS), a commonly used software platform for mobile robots. The sensors on our robot are a light detection and ranging sensor (LIDAR), and a Kinect™. The Kinect can yield depth information using a structured light (SL) sensor, together with a conventional RGB video image [50]. The LIDAR sensors used for cost-constrained indoor robots are 2D – they provide the range vs. angles while sweeping at a constant height above the floor. Range and precision are reasonably good, but the single vertical slice limits the utility in a geometrically intricate environment such as a home or a store. On the other hand, the SL sensor in the Kinect provides a three-dimensional image of the space in front of the sensor. Each pixel is tagged with depth

information, but the range and resolution are limited. In this dissertation we propose an innovative algorithm to combine the strengths of these readily available, inexpensive sensors to generate a useful multilevel map so that mobile robots can perform better in retail environments.

Our mapping algorithm is based on the Rao-Blackwell Particle Filtering (RBPF), which is the most widely used simultaneous location and mapping (SLAM) implementation. We utilize the strengths of the robot's two sensors, the Kinect and LIDAR. Instead of merging both of the sensor's output data, our algorithm can handle the two data streams separately, and manipulate them to provide a multilevel map. We will first discuss the traditional RBPF for the SLAM problem below, and then detail our mapping algorithm.

4.1.1 RBPF for SLAM

SLAM performs the tasks of building a map while estimating the pose (location and orientation) of the robot within a given map. The most widely used SLAM implementation is the Rao-Blackwell Particle Filtering (RBPF) [51]. It has two inputs: odometry and sensor observation, and two outputs: map and pose. It works as expressed by the Rao-Blackwell theorem below:

$$P(x_{1:t}, m | z_{1:t}, u_{0:t-1}) = P(x_{1:t} | z_{1:t}, u_{0:t-1})P(m | x_{1:t}, z_{1:t}) \quad (1)$$

In Eq. (1), x is the pose of the robot, m is the map, z contains the observations, u contains the odometry measurements, and P is conditional probability. The subscripts denote the discrete time, for example, $x_{1:t}$ denotes the poses from time $T=1$ to $T=t$. The key idea of RBPF is to partition the SLAM problem into two sub-problems: pose estimating and map updating. The map will be updated with the estimated pose. In Eq. (1), $P(x_{1:t} | z_{1:t}, u_{0:t-1})$ is called robot path posterior, it represents the pose estimate. This is a typical localization problem, and it can be

solved by the Monte Carlo localization algorithm [52]. $P(m|x_{1:t}, z_{1:t})$ represents the solution to the problem of generating a map when the pose is known. Rao-Blackwell Particle Filtering [53] uses a set of particle filters, each particle with its own trajectory (a trajectory is represented by a sequence of poses $x_0, x_1, x_2, \dots, x_t$) and the related map. The updating step for Rao-Blackwell Particle Filtering is as follows:

- Pose estimating: each particle estimates its current poses x_t from prior poses ($x_0, x_1, x_2, \dots, x_{t-1}$) and observation against the existing map.
- Map updating: each particle updates its own map with estimated pose x_t in previous steps and observations.
- Weighting: Each particle survives with a weight (also called likelihood) of how well the observations match to its own map.

After one updating step is done, estimated pose and mapping can be selected from the best-matched particle.

4.1.2 Multiple-layer Mapping Algorithm with Both LIDAR and Kinect

Let's take a look at the detail of Pose estimating of the RBPF approach. Pose estimating is based on the Monte-Carlo Localization [52], the main steps are as follows:

- Movement updating: Generate from each previous particle to a new particle according to the motion model: $x_t = x_{t-1} + u_{t-1}$.
- Observation updating: weigh the particles with the observations likelihood: $w_t = P(z_t|m, x_t)$.

The estimated pose can be selected from the highest weight particle. In the observation updating step $P(z_t|m, x_t)$ present the measurement probability, which tells how well the observations z_t match up with the map m when robot pose is x_t .

As seen from the above description of pose estimating step, observation is a key factor of SLAM. The more information the observation can provide, the more precisely the pose can be estimated and the more realistically the map can be generated.

Mapping with any single sensor has its limitations. For example, we use a 360-degree Laser Scanner Development Kit (RPLIDAR) [54] in our experiment. However, the drawback of our LIDAR is that it only provides data in a 2D horizontal plane, so it will miss important information about vertical structure. This will increase the likelihood of collision with objects, such as tables, that are non-uniform in the vertical dimension. However, due to its long range and large angle of view, it can yield reliable and precise pose estimation. Figure 4.2 shows the probability density function derived from LIDAR, Kinect, and the odometry. We can see that the observation from LIDAR offers the most precise estimate.

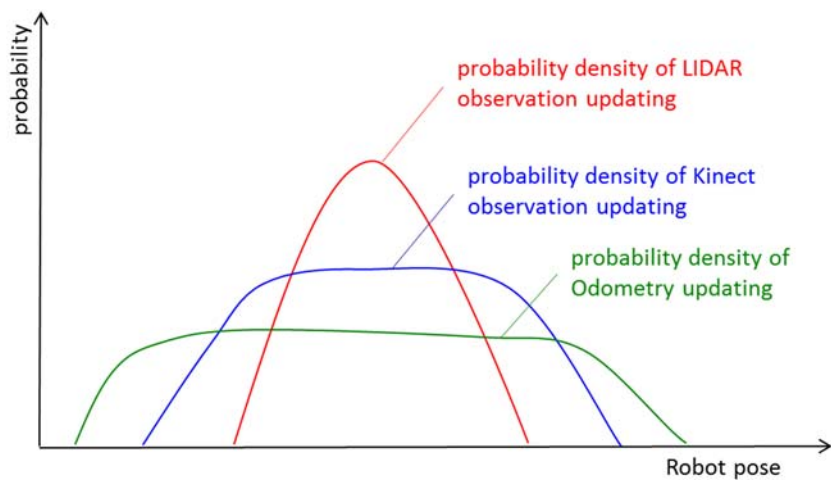


Figure 4.2. Pose probability density functions of LIDAR, Kinect, and the odometry.

On the other hand, the Kinect has shorter range and limited angle of view, but provides 3D depth which represents the environment more completely. As shown in Figure 4.2, the pose estimate from the Kinect has higher uncertainty than from the LIDAR.

Then we try to combine these two sensors together. Because SLAM with LIDAR can provide an precise pose, we first generate a pose from LIDAR and a map called LIDAR-layer-map. We can directly use this precise pose for registration of the Kinect's image. With the depth information from the Kinect and the precise pose, we can get another map called Kinect-layer-map. In this map, we will have a precise pose and all useful information from the environment. In this way, we will get two separate maps but with high consistency in coordinates, which makes up a multiple-layer map. Our multiple-layer mapping algorithm is outlined as below.

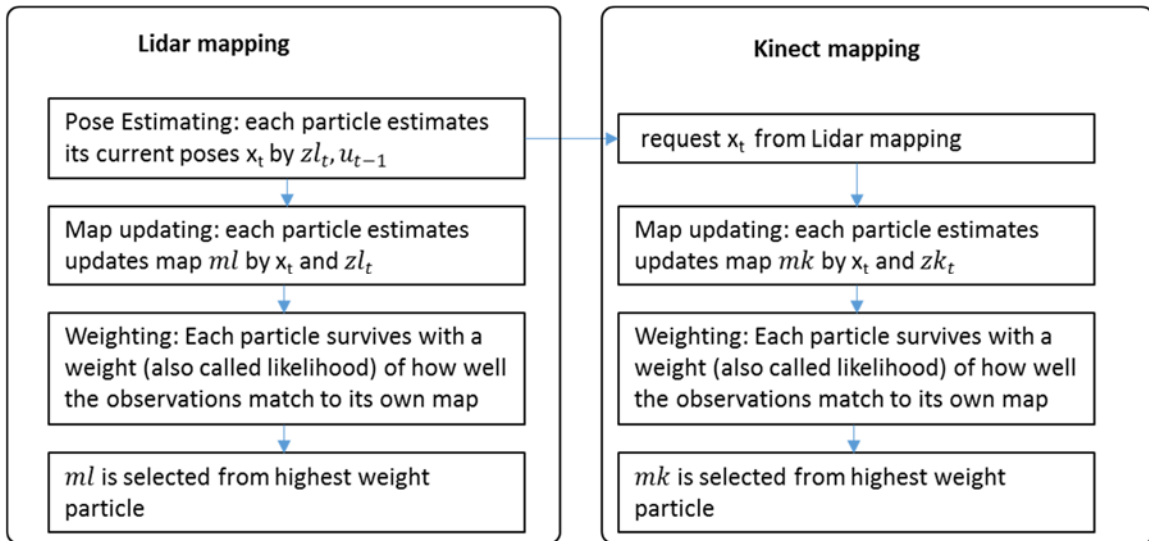


Figure 4.3. Outlines multiple- layer mapping x is the pose of robot, ml is the LIDAR-layer-map, mk is the Kinect-layer-map, z_l contains the observations of the LIDAR, z_k contains the observations of the Kinect, and u contains the odometry measurements.

4.2 Robot Localization

After the multiple-layer mapping algorithm makes an environmental map, the robot can navigate against this map. During the autonomous navigation, the robot must do a very critical task, which is localizing itself against the given map. In order to reduce the computation complexity, we first project the multiple layer map into a one layer map that only has the outline of objects. This map keeps the most important information of objects that are sufficient for robot localization and collision avoidance. In our work, we implement the famous Monte-Carlo Localization [52].

4.3 Inventory Based Path Planner

The inventory based path planner generates a navigational path from the given map. The path should ideally cover all of the merchandise in the map. From the knowledge of the RFID antenna model, the robot is required to move close to the merchandise to get higher read rates. In the costmap used to represent the robot's environment, merchandise-rich areas are represented as occupied cells. They are classified as "obstacles" because they impede the robot's motion. However, unlike obstacles normally encountered in robotics, these have the additional characteristic of being regions of high interest for scanning. The task of the path planner is to find a suitable path in the given map that will cycle around all such "interesting obstacles" in the map enough times to perform a reasonably exhaustive inventory count.

Our inventory based path planner comprises two layers or levels, the global planner and the local planner. The global planner generates the path to cover all the merchandise-rich regions of the map. The path is represented by a group of discrete goals, such that when the robot navigates to all the goals it will have covered all the merchandise. The local path planner determines how to

move from one goal to another, and it also handles the dynamics of the environment, for example a person standing in front of the robot.

4.3.1 Global Path Planner

The global path planner generates a path that covers the entire map while circling around all the merchandise fixtures, ideally enabling the robot-mounted reader to read all the RFID tags in the map. The use of passive RFID tags requires the reader antenna to send enough power to trigger the tags. This requires the global path planner to balance the requirement to get close to the tags with the need to avoid collisions with the store fixtures. The final global path is represented by a group of discrete navigational goals.

The following paragraphs describe in detail how the costmap is filled in. Firstly, the map is rendered in grayscale. Free space cells are assigned the value 0, and occupied space is assigned the value 1. Unknown space is specially marked, for example assigned a -1, which will help to validate the path if we do not want the robot to move into an unknown area. Our aim is to find the best route around the obstacles in the map. This route must be closed, but cannot cause the robot to collide with other obstacles. A brief outline of the different steps of the global path planner algorithm is given in table 4.1. We will discuss details in the later steps.

Input: (X, l) presenting the grayscale map and dilate parameter

Step 1. Dilate the input grayscale map

Step 2. Detect the edges of dilated map

Step 3. Sample the edges as the global path

Step 4. Assign the orientation of each global path goal

Table 4.1. The outline of the global path planner algorithm

Step1. Dilate the input grayscale map

Intuition dictates dilating the obstacles of the map to a specific scale, and choosing the route in the edge of the dilated obstacles. The dilation is represented by the equation below.

$$D = A \oplus B$$

In the above equation, D denotes the image after dilation, A is the original image of the map, and B is the dilating kernel, \oplus denotes the dilation operation. We choose a square kernel to dilate the image. The size of the kernel is controlled by the input parameter l . Changing the size of the kernel will change the size of the dilated obstacles. Since the robot's path will be chosen to lie along the edges of the dilated obstacles, the size of the kernel can control the distance from the reader antenna to the merchandise.

Step 2. Detect the edges of the dilated map

The edges of the dilated map objects must be determined to extract the potential route. In this work, we use the well-known Canny edge detector [55]. This has been implemented in our work using Python, and also by using OpenCV.

Step 3. Sample the edges

After we get the edges of the dilated map, we sample the edges, and choose the sampled edge points as the goals of the path.

Step 4. Assign the orientation of each global path goal

A goal of the global path comprises of two main parts: the position and the orientation in the map. The position of each goal is given by the sampling step. We need this step to assign the orientation to each goal. The strategy is each goal should face the next one. Following this rule, after assigning the orientation to all goals, the navigation of the robot will become smoother.

The orientation of a goal is given by the below equation:

$$\theta_1 = \text{atan2}(g_2.y - g_1.y, g_2.x - g_1.x)$$

Here θ_1 is the orientation of goal g_1 , g_2 is the next goal of g_1 . $(g_1.x, g_1.y)$ and $(g_2.x, g_2.y)$ is the position of goal g_1 and g_2 in the given map.

The following figure shows the various stages of our global path planner.

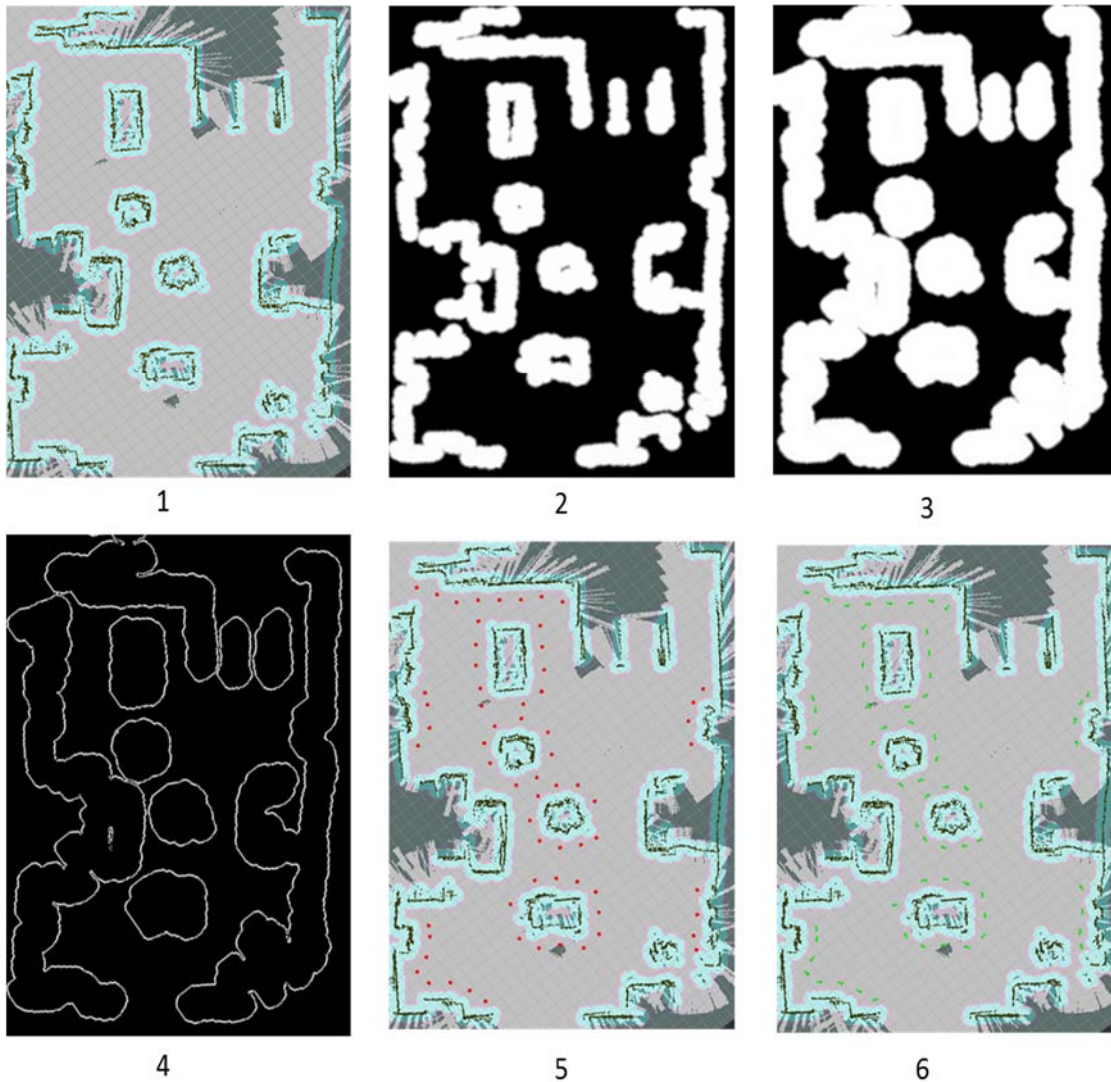


Figure 4.4. The output of the global path planner at various stages. Panel 1: Original map, Panel 2: Original map converted to grayscale, Panel 3: Dilated image, Panel 4: Edges extracted from the dilated image, Panel 5: Sample global path, sampled global goals are indicated by red dots, Panel 6: Orientated Global path, the green arrows indicate the positions and orientations of the global goals.

4.3.2 Local Path Planner

After executing the global path planner, we get a group of goal points that represent the global path. However, the robot still needs to figure out how to move among these points, and it has to handle the dynamics of the environment, like a person moving into the path of the robot.

Our local navigational path planner is based on the Dynamic Window Approach (DWA) [56]. It will figure out the route from one global goal to another, and efficiently respond to the unforeseen dynamics of the retail environment. Our local planner is especially designed to deal with the constraints imposed by limited velocities and accelerations for a short time interval. To ensure the efficient response to the highly dynamic environment, the local planner takes the limitation of the sensors into account. Trajectories are approximated by circular curvatures, resulting in a two-dimensional search space of translational and rotational velocities.

Translational velocity is denoted by v , rotational velocity by w . A brief outline for the one cycle of our local planner is given in Table 4.2.

<p>Step 1. Generate Circular trajectories : Generating a set of circular trajectories curvatures which are uniquely determined by pairs (v,w) of translational and rotational velocities. This results in a two-dimensional search space of translational and rotational velocities.</p> <p>Step 2. Admissible velocities: Executing the restriction of Admissible velocities into trajectories, which are generated by step 1, only safe trajectories are considered. A pair (v,w) is considered admissible if the robot is able to stop before it collides with the closest obstacle.</p> <p>Step 3. Dynamic window: The dynamic window restricts the trajectories of step 2, only the trajectories whose velocities can be reached within a short time interval are considered.</p> <p>Step 4. Sensor limitation: Filtering out the trajectories that violate the sensor limitation.</p> <p>Step 5. Optimization: In this step the trajectories maximizing an objective function that is chosen from the remaining trajectories.</p>
--

Table 4.2. Outline of local path planner

Step 1. Generate Circular trajectories

Reference [52] shows that the trajectory of a synchro-drive robot can be approximated by a sequence of circular arcs, which have circular curvature. A curvature is uniquely determined by the velocity (v,w) of translational and rotational velocities. This is the first step of the local path planner, it will generate a set of trajectories to a given goal point for a very short time intervals ∇T . We assume that the velocities within ∇T are constant, because usually the ∇T is very short. With this assumption, we can make the search space two-dimensional and thus tractable.

Step 2. Admissible velocities

In this step, we will filter out the trajectories which may cause the robot to collide into obstacles. We can check the trajectories within the ∇T to make sure that the robot will not collide with any obstacles. Furthermore, a trajectory is considered admissible if the robot moves with corresponding velocity, and is able to stop before it reaches this obstacle.

Step 3. Dynamic window

In this step, we will take into account the limitation of the robot's acceleration which is restricted by its motors. Only the trajectories with velocities that can be reached within a time interval given the limited accelerations of the robot are considered.

Step 4. Sensor limitation

This step is critical, it enables the robot to safely navigate in the retail environment. It will take into account the sensor limitation. A retail environment is highly dynamic and unstructured, for example, it may contain uneven objects and people may suddenly move close to the robot. The local planner must handle these obstacles efficiently to avoid colliding into them. However, every sensor has its own limitations, which sometimes causes the robot to fail to fully detect its

surrounding environment. Consider the Kinect for example; it has a narrow viewing angle, 43 degrees in vertical, 57 degrees in horizontal, 0.8 to a 4 meter long range. The Kinect can detect the objects only in this range. In order to detect all the dynamic objects, the local planner only considers the trajectories that navigate the robot within the sensor’s detectable range. The idea of this step is illustrated by the figure 4.5 below.

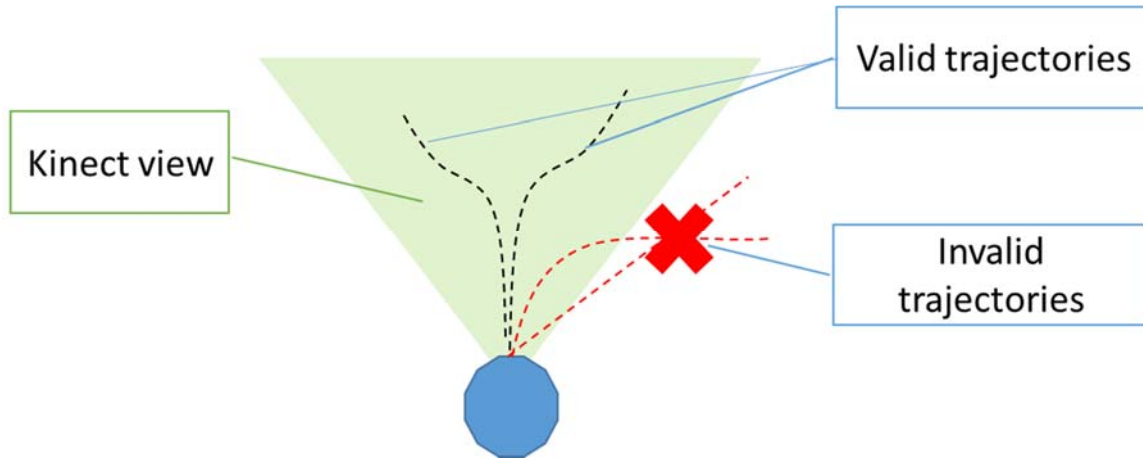


Figure 4.5. Sensor limitation, here we assume the Kinect is the only sensor of the robot. The blue circle represents the robot. Only the trajectories that guarantee the robot’s navigation within the Kinect’s view is valid. We consider trajectories within a very short time interval.

Step 5. Optimization

The local planner chooses an optimal path in this search space for the robot to move from one goal to another. The optimal path will maximize the objective function, which is the same as traditional DWA:

$$G(v, w) = \delta(\alpha \cdot heading(v, w) + \beta \cdot dist(v, w) + \gamma \cdot vel(v, w)) \quad (1)$$

- *Heading*(v, w) is a measure of progress towards the goal location.
- *dist*(v, w) is the distance to the closest obstacle on the trajectory.
- *vel*(v, w) is the forward velocity of the robot.

- The function δ smoothes the weighted sum of the three components and results in more side clearance from obstacles. α, β and γ are adjustable parameters for choosing an optimal trajectory.

In order to maximize received signal strength, the robot needs to move as close as possible to the merchandise, so the local planner should choose a correspondingly suitable route. An example to demonstrate this idea is shown in Figure. 4.6.

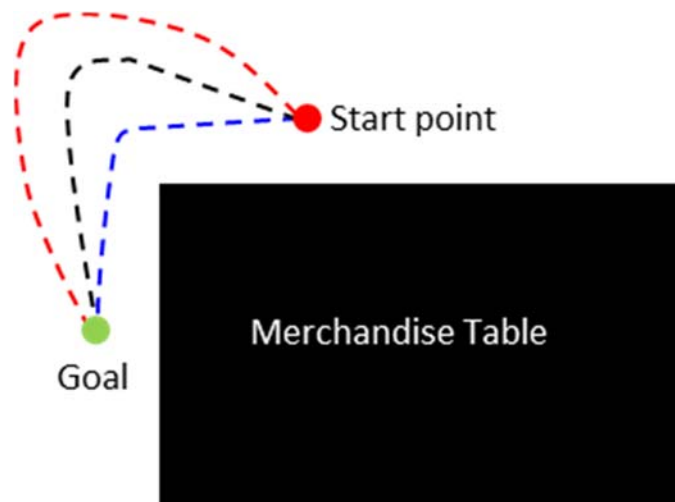


Figure 4.6. The local planner will generate multiple routes, and RFID inventory quality is maximized by the one closest to the merchandise.

In Figure. 4.6, the robot needs to move from the starting point to the goal, both of which are near the merchandise table. The DWA will generate multiple candidate routes, as shown by dashed lines in the figure. The best route for RFID inventory is the blue one, which will take the robot close to the table. So, the value of β is set larger than normal to make the robot always choose a route close to obstacles as an optimal trajectory.

Chapter 5 RFID Tag Localization

The key factor of inaccurate inventory is misplaced items. It is said that 16% of the items in the stores of main retailers in United States are not located in the correct places [57]. Being able to localize RFID tags can dramatically increase the chance to find misplaced products. It demonstrated by [58] that localization technologies can significantly reduce the time to retrieve the misplaced items in a warehouse. As the use of RFID tags increases, the benefits of localizing RFID tags, especially passive RFID tags, is also increasing.

However, passive RFID is not easy to localize precisely. Firstly, most of the commercially-used RFID tags cannot provide calibrated RSSI information; this prevents us from using RSSI as an observation for localization. Secondly, as we have introduced in Chapter 3, the retail environment is highly unstructured, and that induces radio wave reflection (multipath fading), which is the greatest complicating factor to affect location accuracy. Multipath fading is so heavy in retail indoor environment that time-of-arrival algorithms such as LTOA cannot yield an acceptable localization result.

In this chapter, we will introduce two RFID tag localization algorithms, (1) fixed power and (2) variable power tag localization. The goal of those algorithms is to accurately report the location of multiple passive RFID tags in a pre-determined area. Both algorithms are probabilistic approaches based on recursive Bayesian updating. They both have obvious advantages and disadvantages, and can suit for different scenarios. They are both collecting the output of the conventional RFID reader, combined with robot pose, and use this information with

a probabilistic antenna model to estimate the location of RFID tags. We will first introduce the recursive Bayesian updating algorithm, then present the method of building a probabilistic antenna model, and detail the two RFID tag localization algorithms.

5.1 Bayesian Updating

Bayesian updating or Bayes filter is a widely researched topic in the robotics community, and addresses the problem of estimating belief over the state x of a dynamical system for sensor observations. The Bayes filter is recursively updating the belief $bel(x_t)$ at time t , and $bel(x_t)$ is calculated from the belief $bel(x_{t-1})$ at time $t-1$. In this section, we will firstly introduce the typical Bayesian updating for robot state estimating, and then illustrate how to modify it for RFID tag localization. Table 5.1 provides pseudo-code for a step of a typical Bayesian updating.

<ol style="list-style-type: none"> 1. Bayesian updating ($bel(x_{t-1}), u_t, z_t$): 2. for all x_{t-1} do: 3. $\overline{bel}(x_t) = \int P(x_t u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$ 4. $bel(x_t) = \eta\overline{bel}(x_t) P(z_t x_t)$ 5. end for 6. return $bel(x_t)$
--

Table 5.1. Pseudo-code of one step of Bayesian updating

The Bayesian updating algorithm calculates belief distribution $bel(x_t)$ recursively. It calculates the $bel(x_t)$ from control u_t , observation z_t and prior belief $bel(x_{t-1})$, which is calculated previously.

The typical Bayesian updating algorithm has two essential steps. The first step is called the control update or prediction. The main idea of control update is illustrated in Line 3 of Table

5.1. It calculates a belief of state x_t based on the prior belief of state x_{t-1} and the control u_t . The belief $\overline{bel}(x_t)$ represents the belief distribution of a robot to make a transition from state x_{t-1} to x_t , which under the condition of the given control u_t and the prior belief $bel(x_{t-1})$.

The second step of Bayesian updating is called the measurement update. The main idea of measurement update is illustrated in Line 4 of Table 5.1. $P(z_t | x_t)$ is the probability of robot in state x_t to observe the measurement of z_t . The $bel(x_t)$ is the product of $\overline{bel}(x_t)$ and $P(z_t | x_t)$. It does so for every hypothetical posterior state x_t . The η is a constant, which helps to integrate the sum of all $bel(x_t)$ into 1.

To compute the $bel(x_t)$ recursively, we should provide an initial belief $bel(x_0)$ at time $t=0$. Usually the initial belief can be given; for example, we know the start pose of a robot. If the initial belief is unknown we can give normalized probability to all hypothetical states, after several Bayesian updating steps the states will be converged.

Let's take look at how to implement Bayesian updating for RFID tag location estimating in a retail environment. The first assumption is that during the inventory period the RFID tags are stationary. This assumption is reasonable, because during most of time the merchandise in retail is stationary. Reference [7] introduces a Bayesian updating algorithm for HF RFID localization. Let x denote a two-dimensional vector that represents the location of a RFID tag. z_t denotes a binary variable that represents the observation of RFID reader against a RFID tag at time $T = t$. h_t represents the location of reader at time $T = t$. The $bel(x_t)$ in Table 5.1 can be denoted by $P(x|z_t, h_t)$. Because the RFID tags are stationary, in the control update step:

$$\overline{bel}(x_t) = \overline{P}(x|z_t, h_t) = P(x|z_{t-1}, h_{t-1}) \quad (1)$$

Let's substitute the equation (1) into Line 4 of Table 5.1,

$$bel(x_t) = P(x|z_t, h_t) = \eta P(x|z_{t-1}, h_{t-1}) P(z_t | x, h_t) \quad (2)$$

In equation (2) $P(z_t | x, h_t)$ is called a RFID model, which gives the probability of a tag, which is located in x , to be observed by a RFID reader which is located in h_t . We will detail how to build a RFID model in the next section. We can see from equation (2), if we can measure the observations of a RFID tag in several locations, the RFID tag location can be estimated by recursively updating the observations.

5.2 Antenna Model

As illustrated by equation (2), the Bayesian updating algorithm highly relies on the RFID model $P(z_t | x, h_t)$. One could develop a RFID model as a function of the relative distance between the tag and the reader antenna. Theoretically, a propagation model can be applied to build a RFID Model. In unobstructed free space, Friis transmission equation [59] shows that the signal strength level decreases at a rate inversely proportional to the distance travelled,

$$P_r = P_t \frac{G_t G_r \lambda^2}{16L\pi^2 d^2} \quad (3)$$

In equation (3) P_r is the power received by RFID tag antenna and P_t is the power input to RFID reader antenna. G_t is the reader antenna gain, G_r is the RFID tag antenna gain and L is the system loss factor. λ is the wavelength, and d is the distance between the reader antenna and a RFID tag. Based on this relationship we can build a map that indicates tag received signal strength. We can convert this map into a RFID model $P(z_t | x, h_t)$ by an intuitive rule that is higher signal strength to higher observed probability. However, this method is only feasible for simple geometries, and therefore it is not practical for retail environments.

Instead, we built our model through experiments. We mounted a RFID reader antenna into our robot, which is the same layout as we use for the RFID localization. And we placed a RFID tag at 10 cm intervals around the reader antenna while recording the numbers of successful observations by reader after a fixed time period, like 2 seconds. We repeated our experiment 20 times, then we created a map of numbers of successful observations. We converted this map into RFID model $P(z_t | x, h_t)$, and higher successful observations gained higher observed probability. Figure 5.1 is an example of our experiment based RFID model. The experiment reader is Zebra FX9500 Fixed RFID Reader and the antenna is Zebra AN720 RFID Antenna. And the interval of the RFID model is 10 cm.

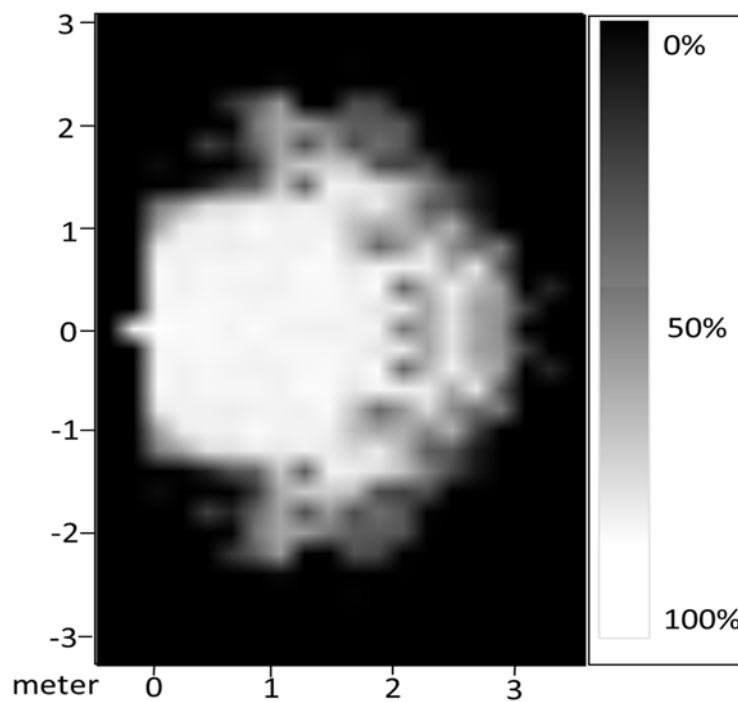


Figure 5.1 RFID model, the RFID reader antenna is setting in position (0,0). Here, the reader is Zebra FX9500 Fixed RFID Reader and the antenna is Zebra AN720 RFID Antenna. The interval of the model is 10 cm.

5.3 Fixed Power Tag Localization Algorithm

The fixed power tag localization algorithm can be divided into two phases. The first phase is called observation collecting. In this phase, a robot mounted with an RFID reader and antennae will slowly navigate in the target area. The robot collects the RFID tag responses while navigating. Every record of the RFID tag response will be associated with the reading antenna pose(x, y, Θ) against a given map. During whole observation collecting, the reader will work under a fixed transmission power setting. In order to get maximum numbers of RFID tag responses, the reader usually works under the maximum transmission power setting.

The second phase is the RFID tag location estimating. After the robot navigates the target area, the robot has collected all the RFID tag responses in varied locations. Then we use recursive Bayesian updating combine the related RFID model to estimate the locations of RFID tags. The next table illustrates the main steps of estimating a RFID tag location by fixed power tag localization algorithm.

<p>Input: all observed records of a RFID tag, the records include related antenna poses</p> <p>Step 1. Initialize hypothetical posterior states or locations x, we assign all possible locations of the tag. Assign normalized belief to every $bel(x_0)$.</p> <p>Step 2. For every observed record updating the $bel(x_t)$ by equation (2).</p> <p>Step 3. Check that the tag is localizable or not, if yes, find the highest belief in all hypothetical locations, and this location is the estimated location for the tag.</p>
--

Table 5.2. The tag location estimating of fixed power tag localization algorithm

In step 1, the hypothetical posterior states or locations x can be initialized by all possible locations. The possible locations are generated from all the observed records of a RFID tag. We find the boundary of the observed records, which beyond this boundary the reader does not read

any response of the tag. All the positions (x', y') inside of this boundary are the hypothetical posterior positions. In order to reduce the computing effort, we sample these positions, for example, the positions can be sampled at the same interval of the RFID model. Then the discrete points $x \in \{(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_n, y'_n)\}$ are the initialized hypothetical posterior positions. Then we will initialize an even belief $bel(x_0)$ for every potential position, for example $bel(x_0) = 1$.

In step 2, we will update the belief $bel(x_t)$ in every hypothetical posterior position for all observations by equation (2). In equation (2), for a given potential position $x = (x'_n, y'_n)$ the probability of the measurement can be given by RFID model $P(z_t | x, h_t)$.

In step 3, after updating all the observations, we find the $bel(x_t)$ will converge, that is only a small number of hypothetical posterior positions x will survive. A hypothetical posterior position survives when the $bel(x_t)$ greater than a threshold, for example 0. The less surviving potential positions the better confidence of the tag localization. We can introduce a threshold T_{su} of the number of surviving hypothetical posterior positions as a criterion. When the surviving hypothetical posterior positions are greater than T_{su} we can tell this tag is not localizable, because we may failed to collect enough observations for it. Then we can find the highest belief in all surviving hypothetical locations, and this location is the estimated location for the tag. The following figure illustrates the basic idea of every step of the fixed power tag localization algorithm.

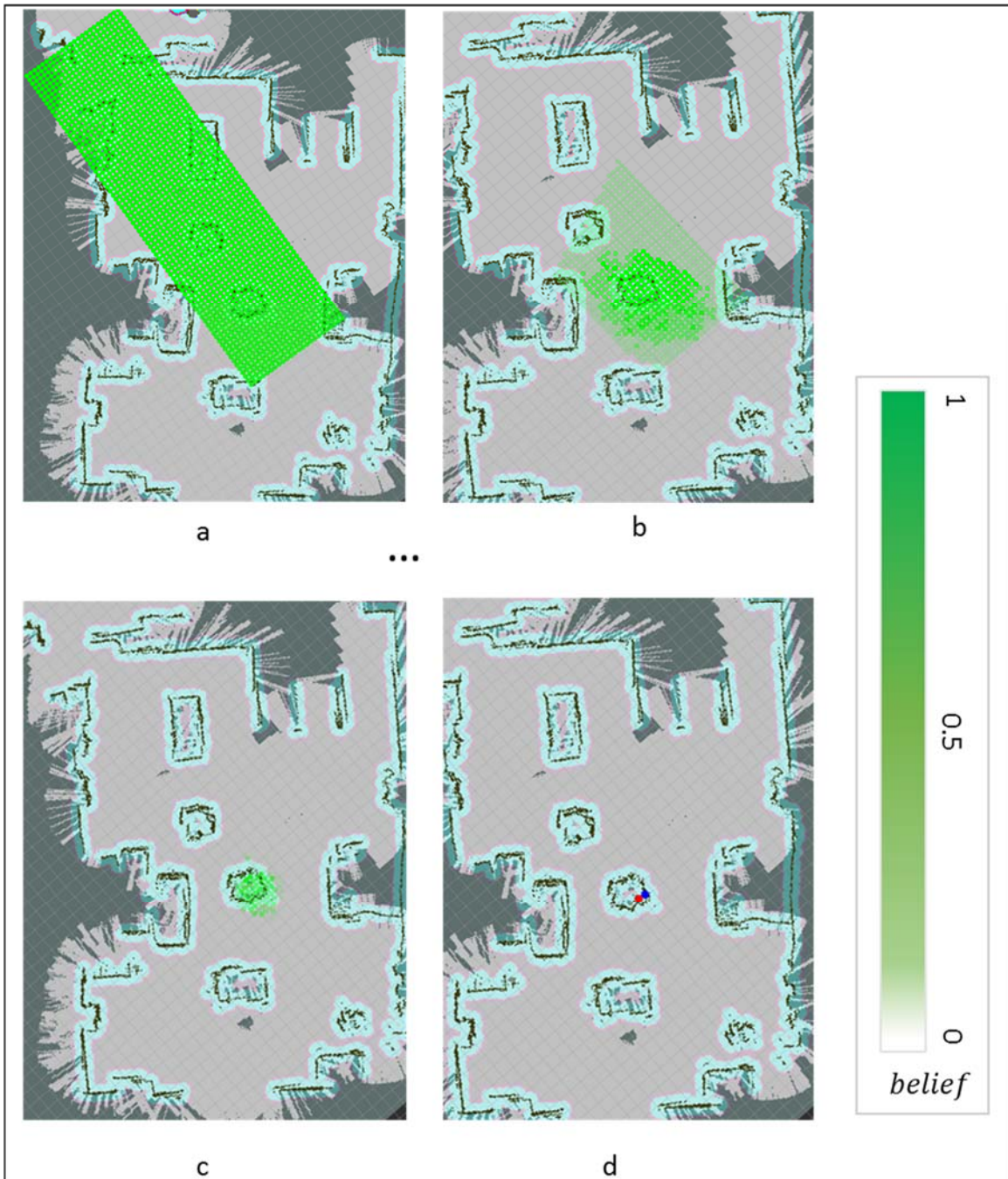


Figure 5.2. Procedures of fixed power tag localization algorithm, (a) initial hypothetical posterior locations x , (b) $bel(x_i)$ after first observation updating, (c) $bel(x_i)$ after all observations are updated, (d) location result: the red point is the estimated location of this tag, and the blue one is the ground truth position, and error in this case is 0.28 meter.

5.4 Variable Power Tag Localization Algorithm

Passive RFID is not a precise location technology. The reader model only can provide a rough location estimation. Consider Figure 5.1 as an example. If the reader detects an RFID tag, it only can tell the tag may exist in a 4 square meter area. In the fixed power tag localization algorithm, the low-resolution RFID model $P(z_t | x, h_t)$, is one of the main factors to affect location accuracy. If we can improve the resolution of the RFID model, we can greatly improve the accuracy of the algorithm.

One approach to improve resolution of the RFID model is to reduce the transmission power of the RFID reader. Take figure 5.3 as an example, the resolution of the model under power 80 is less than 1 square meter, by contrast the resolution of the model under power 170 is more than 4 square meters. But the drawback of reducing the transmission power is that the detectable range will be reduced dramatically, in figure 5.3 the detectable range reduced from 3 meters under power 170 to 1 meter under power 80. This drawback is not acceptable for a robot working in retail environment, because most of the merchandise items are located farther than the 1 meter of the robot's navigation trajectory.

We propose a method that can improve resolution of the RFID Model while keeping the detection range. While collecting RFID tag responses in a position, we vary the transmission power of the RFID reader. By deploying variable transmission power, we can greatly improve the resolution of the RFID model. For example, when a tag can be detected at a high power level, but cannot be detected at a low power level, we can tell that the tag has high probability of existing in the area that is inside the high power detectable area and outside of the low power detectable area. After collecting all the responses under different transmission powers, we combine the measurements of each individual transmission power to create a vector-based

measurement. Then using this vector measurement to estimate the location of tags. This approach is called the variable power tag localization algorithm.

The variable power tag localization algorithm also can be divided into two phases, observation collecting phase and RFID tag location estimating phase. In the observation collecting phase, the robot needs to stop in several positions to collect the RFID tag responses. These stopping locations are called measurement points. In each measurement point the RFID reader scans tags at several power levels and records the results for each individual tag. For each tag a reading vector is generated based on the reader antenna pose and the reader powers. For example, a reader will scan at 4 power level indices: 80,110,140, and 170. We assume a tag was scanned only at power levels 140 and 170 from the same antenna, the measurement vector is $[0, 0, 1, 1]$, and we denote this vector as z . A valid vector based measurement consists of this vector and the related reading antenna pose (x, y, Θ) , and we denote this pose as h .

In the RFID tag location estimating phase, based on these vector based measurements, which are collected in the observation collecting phase, a probabilistic localization algorithm is deployed to estimate a RFID tag location. The procedure of RFID tag location estimating algorithm is the same with fixed power tag localization algorithm, which is illustrated in table 5.2. The first step is initializing hypothetical posterior location x , the second step is updating all the observed records by equation (2), then choosing highest belief hypothetical location x as the estimated location. Here, the observation measurement z is a vector that denotes the observation states under varied power levels. The RFID model $P(z_t | x, h_t)$ based variable power vector, provides higher resolution.

The variable power RFID model $P(z_t | x, h_t)$ is generated from fixed power ones. We denote a fixed power RFID model as $P(z_{tl} | x, h_t)$ that is under a certain power level p_l . Since the measurement vector z_t consists of multiple power levels, the $P(z_t | x, h_t)$ can be given by following equation,

$$P(z_t | x, h_t) = \prod_{l=0}^N a P(z_{tl} | x, h_t) \quad (4)$$

Where N is the total power levels, a is the weight. The lower power should have a higher weight, because the lower power levels have lower probability to be affected by multipath fading. The following example shows how to generate a variable power RFID model. It also demonstrates the improvement of model resolution. The RFID reader works under 4 power levels, 170,140,110,and 80. The RFID model for individual 170,140,110 and 80 power levels are shown in the following figure, we use the same reader and antenna as pervious, which the reader is Zebra FX9500 Fixed RFID Reader and the antenna is Zebra AN720 RFID Antenna.

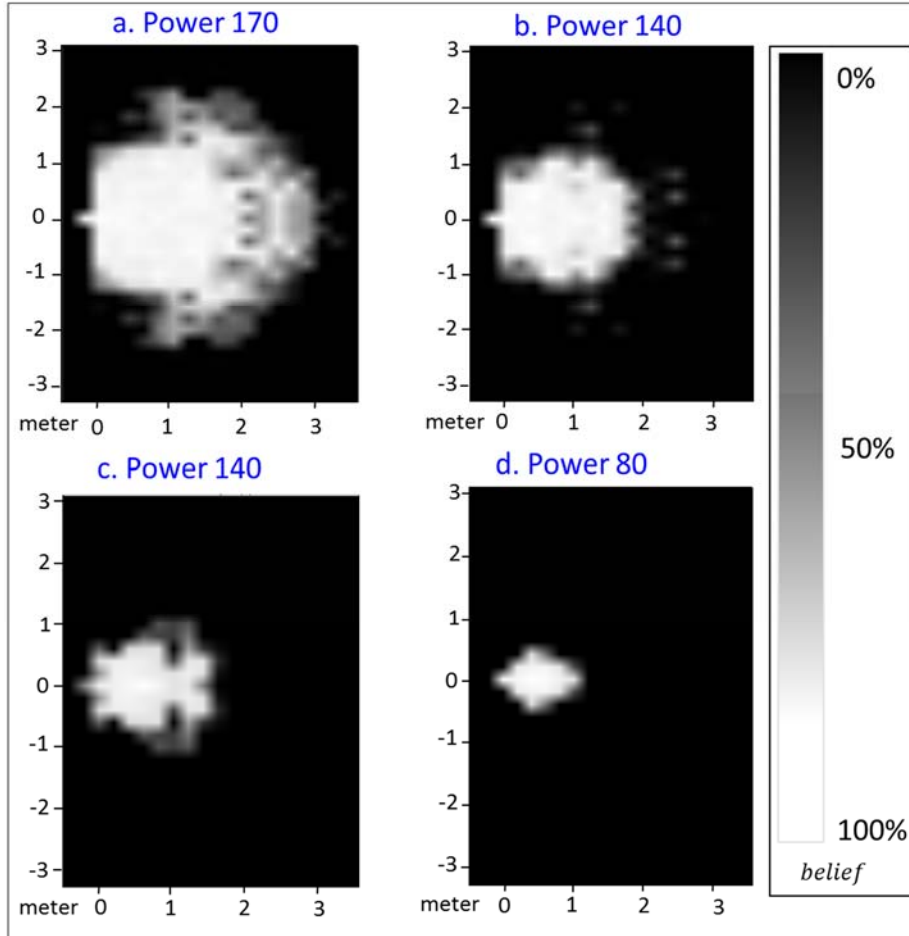


Figure 5.3. The RFID model for Power lever 170, 140,110 and 80. The antenna is set in position (0,0) for every model. Here, the reader is Zebra FX9500 Fixed RFID Reader and the antenna is Zebra AN720 RFID Antenna.

Let us generate variable power RFID model $P(z_t | x, h_t)$ for above power setting by equation (4). The weight for power 170 is 0.1, power 140 is 0.2, power 110 is 0.3 and power 80 is 0.4. The variable power RFID model $P(z_t | x, h_t)$ is shown in the following figure,

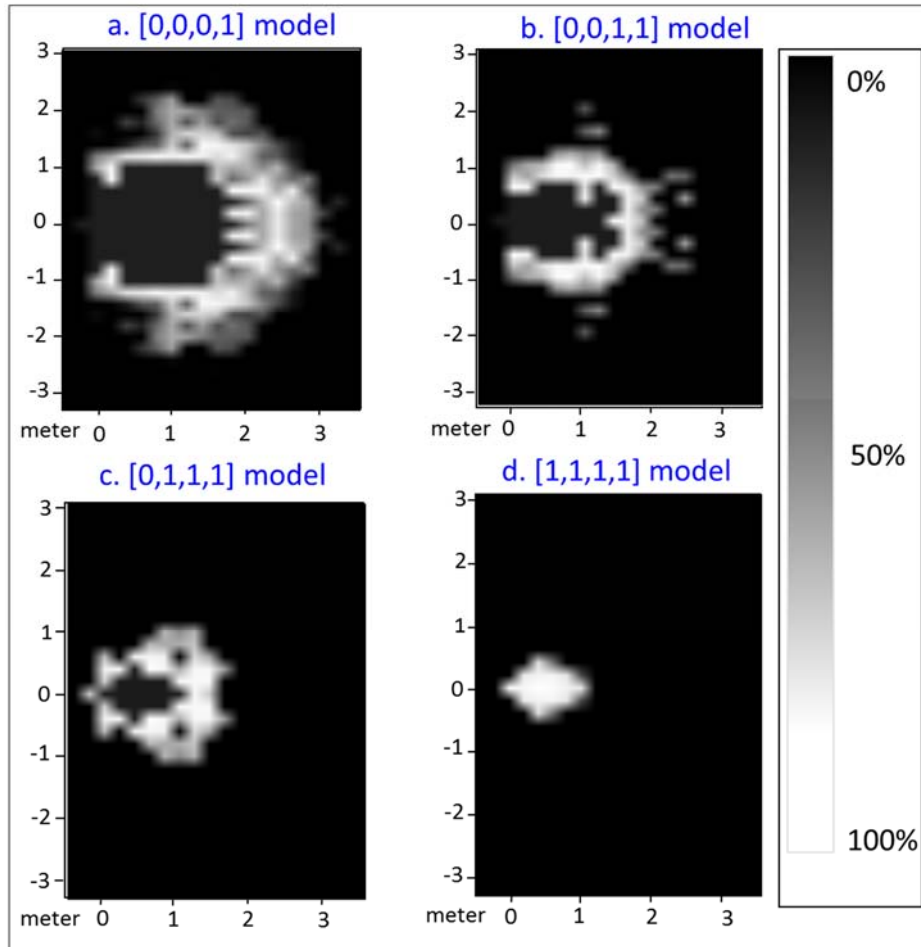


Figure 5.4. Variable power RFID model, (a) the model of measurement vector $[0,0,0,1]$, (b) the model of measurement vector $[0,0,1,1]$, (c) the model of measurement vector $[0,1,1,1]$ and (d) the model of measurement vector $[1,1,1,1]$. The power level setting is power 80,110,140 and 170. The antenna is set in position (0,0) for every model. Here, the reader is Zebra FX9500 Fixed RFID Reader and the antenna is Zebra AN720 RFID Antenna.

From the above figure, we can tell that the variable power RFID model provides higher resolution and keeps the same detect range as the max power level. The valid measurements of the above example are $[0,0,0,1]$, $[0,0,1,1]$, $[0,1,1,1]$ and $[1,1,1,1]$. We will filter out the measurement that get does not show at higher power level but be detected under lower power level, like measurement vector $[0,1,0,0]$ that a tag is detected under power level 110, but can not

be detected by higher power levels 140 and 170. These phenomena mostly are caused by multipath fading of radio waves, which under the higher power level the reflection is stronger that reduce the transmission power in the tag's position because different phase of reflected waves. Under the strong multipath fading phenomena the measurement is not consistent with the RFID model any more. Because a tag may be detected by a reader that is farther than detectable range or cannot be detected even if it is located within the detectable range. We can safely filter out these measurements to improve the estimating accuracy.

Compared to fixed power tag localization algorithm, the variable power tag localization algorithm could provide higher localization accuracy. The drawback of variable power tag localization algorithm is that during the observation collecting phase, it takes a longer time than fixed power does. Because it needs to stop in several measurement points and scan RFID tags in varied power levels.

Chapter 6 Experimental Setup and Results

In this chapter, we present and discuss the experimental results of an autonomous robot that does RFID based inventory and localizes RFID tags in a retail environment. Here, we evaluate the performance of the algorithms developed, including the mapping algorithm, the navigation algorithm and the RFID tag localization algorithm. We perform our experiments in a mock apparel store, and our experiments have the following components:

1. A map is created for the mock apparel store,
2. The robot autonomously navigates in the store,
3. The robot does inventory in the store,
4. RFID tags are localized in the store.

6.1 Experimental Setup

6.1.1 Experiment Environment

To demonstrate the performance of our robotic inventory approach, we conducted experiments using a mock apparel store at the RFID Laboratory at Auburn University. This is an enclosed region laid out with fixtures and products to closely emulate a retail clothing store. The size of the store is 17×12 meters (204 square meters). There are 674 items of merchandise with RFID tags in this room. This total is comprised of 415 jeans and khakis, 149 T-shirts, 33 pairs of shoes, 21 dresses and tops, 30 pieces of lingerie, and 26 pairs of pants. The jeans are distributed in a five layer metal shelf and a five layer wooden shelf. The metal shelf is denoted as number 1 and the wooden shelf is denoted as number 6 in figure 6.1. The shoes are placed in a 3 layer plastic shelf that hangs on a wall, which is denoted as number 8 in figure 6.1. Lingerie is placed

on a wooden table, which is donated as number 2 in figure 6.1. T-shirts, pants and dress tops are hung on 4 metal racks, which are denoted as number 3, 4, 5 and 7 respectively in figure 6.1.

Dresses are hung on 2 metal racks that are denoted as number 9 and 10 in figure 6.1.

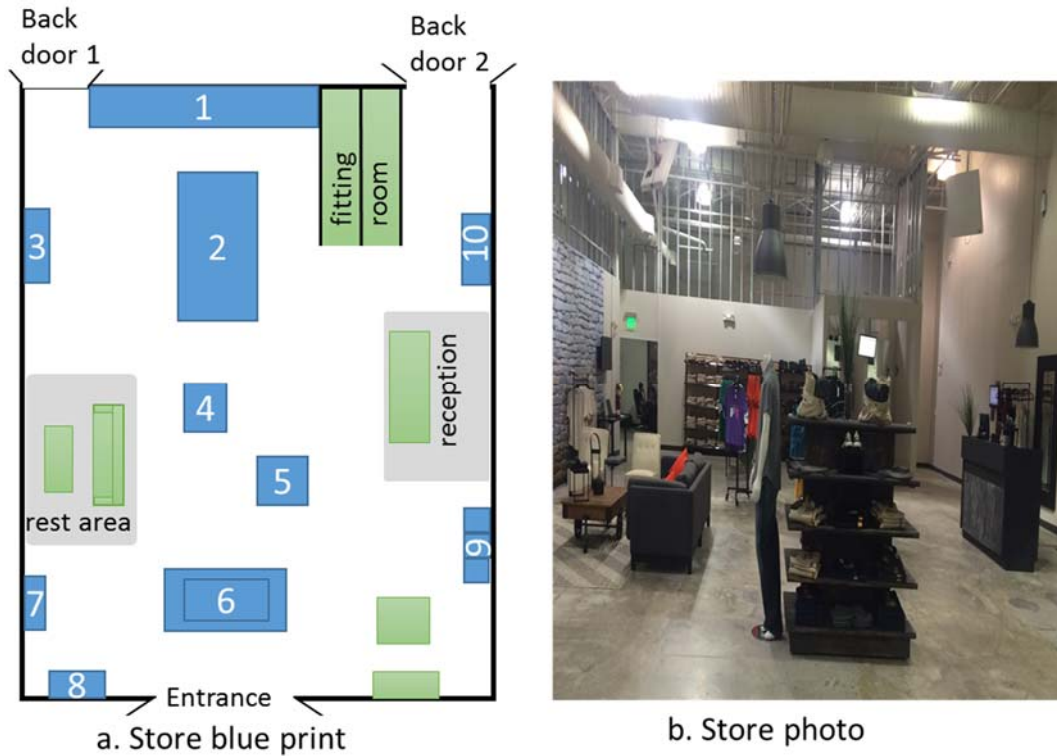


Figure 6.1(a) the top view blueprint map, and (b) photo of the store. The RFID tagged merchandise is distributed in the blue area of blueprint map.

Our experimental RFID tag is Avery 237, which is widely used in retail. All the tags are tagged the same as in a commercial setting. The following photo illustrates a typical tag position on an item.

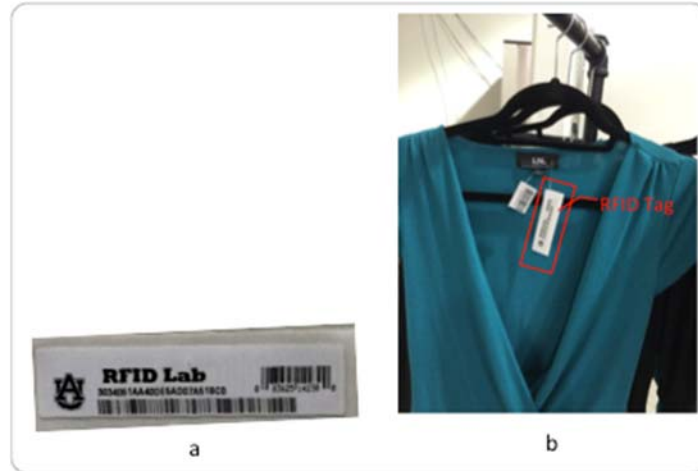


Figure 6.2(a) Experimental RFID tag Avery 237, and (b)The RFID tag attached to a merchandise item

In effect, we establish the experimental environment to mimic a real retail store as much as possible.

6.1.2 Robot System

Our experimental robotic system consists of three parts, an autonomous robot, a wireless network and a remote workstation.

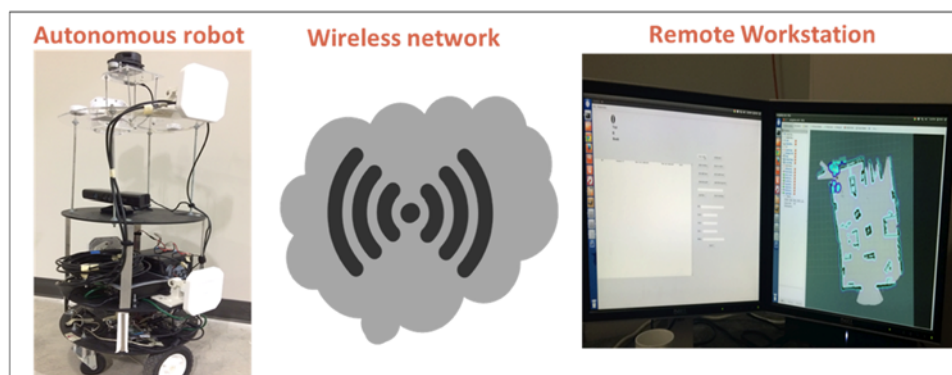


Figure 6.3. The main parts of our experimental robotic system.

Autonomous robot

Our robot is built on a REX-16D Round Robot Base from Zagros Robotics, which consists of two drive motors, two free rotating caster wheels, and three 14-inch diameter ABS

plastic shelves for electronics and payload. One additional disk is added upon the top of the chassis to mount the RFID antennae. The robot components are shown in Figure 6.4.

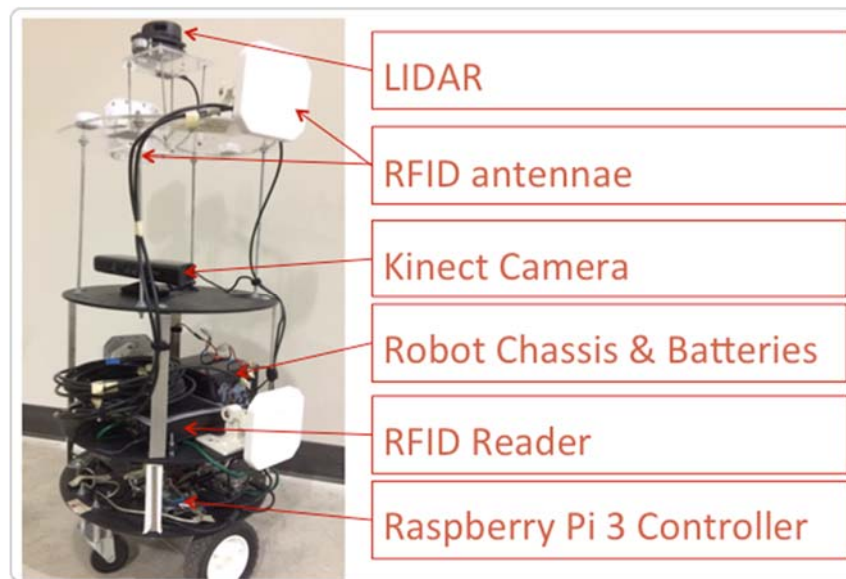


Figure 6.4. Robot components

1) Raspberry Pi 3 controller

We use two Raspberry Pi 3 controllers as the local controllers. The two Raspberry Pi 3 controllers work to process the sensors' input and to navigate the robot in the retail environment. The Raspberry Pi 3 is a powerful credit-card sized single board computer. It provides a 1.2GHz Quad-Core ARM Cortex-A53 Processor and 1GB memory. It provides a USB interface to an Arduino Mega 2560 which performs low-level motor interface tasks, such as PWM output and wheel encoder input. Its built-in Wi-Fi LAN provides efficient wireless communication capability. Both Raspberry Pi 3 controllers use Ubuntu as the operating system, and the software we use is Robot Operating System (ROS). ROS provides open-source libraries and tools to help software developers create robotic applications.

2) LIDAR

For higher elevation scans, we use the RPLidar A1M1-R3 laser scanner. This sensor can perform a 360-degree scan with a maximum range of 6 meters. The scanning frequency used is 5.5 Hz, although it can be configured to a maximum of 10 Hz.

3) Kinect

For 3D environment scans, we use Microsoft Kinect™. This sensor has a 43-degree vertical and 57-degree horizontal field of view. It provides up to 30 frames per second of color and depth images.

4) RFID reader and Antennas

We use a Zebra FX7500 RFID reader and four Zebra AN720 Antennae to read the RFID tags. The Zebra FX7500 RFID reader provides frequency bands of 902 MHz ~ 928 MHz and 865 MHz ~ 868 MHz. It also provides up to 8 antenna ports. The Zebra AN720 Antenna has a frequency band from 902 MHz to 928 MHz and maximum 6 dBi antenna gain. The Beam width of the Zebra AN720 is 100 degrees.

6.2 Mapping Results

To demonstrate the performance of our mapping algorithm, we conducted an experiment in our mock store, which was introduced previously. During the experiment we drove the robot by keyboard in the store, meanwhile the robot built the map for the surrounding environment. The robot built three maps simultaneously, a map built by LIDAR scans only, a map built by Kinect scans only and a map built by our multiple-layer mapping algorithm with both LIDAR and Kinect scans.

Firstly, take a look at the process of projecting a 3D Kinect depth image to a 2D scan while still keeping the important vertical outline structure information. Figure 6.5(a) shows a photo of a real shoe rack. As shown in Figure 6.5(b), the depth image of the Kinect provides vertical structure

information of a shoe rack that is comparable with the RGB photo shown in Figure 6.5(a). After projecting the 3D depth image into the 2D scan, it will contain all the profile information of the shoe rack, as described by the red curves in Figure. 6.5(b).



Figure 6.5(a) photo of shoe rack, (b) depth image of shoe rack

We implemented a standard SLAM in ROS [60] to build a map with only LIDAR scans or only Kinect scans. The map built by LIDAR only is shown in figure 6.6(a). This map is precise, but sparse. Compared with the blue print map in Figure 6.1(a), many items are lost, such as the rest area and the shoe rack. This is caused by the LIDAR only being able to provide data in a 2D horizontal plane. So, it will lose important information about vertical structure, such as rest area and shoe rack, which cannot be detected by the LIDAR at all.

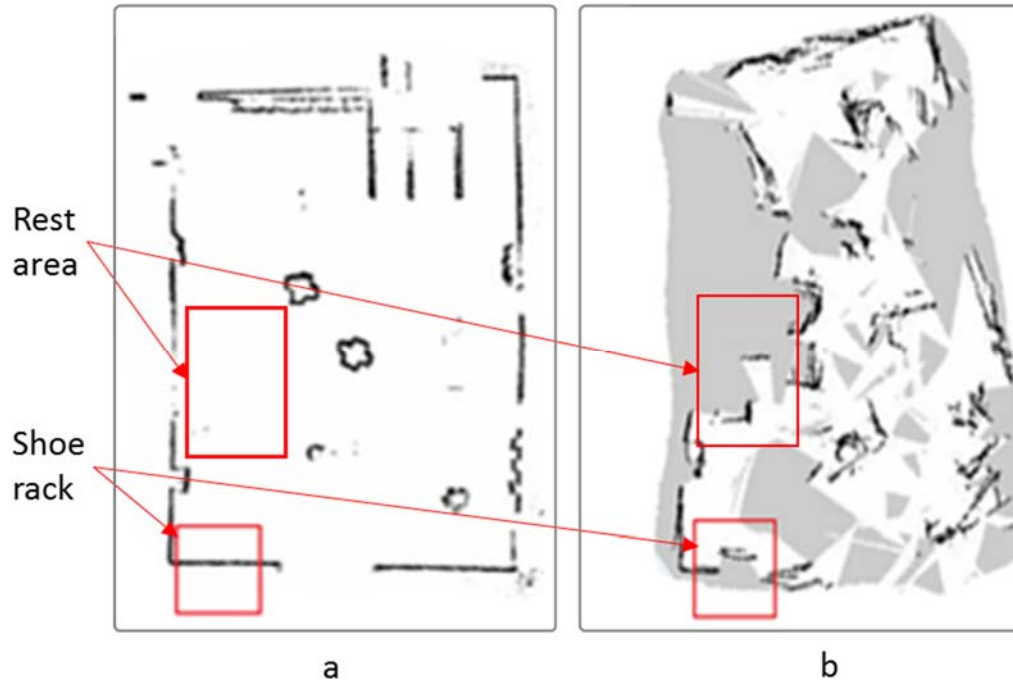


Figure 6.6(a) SLAM with LIDAR scans only, (b) SLAM with Kinect scans only

On the other hand, the map with Kinect scans only is shown in Figure 6.6(b), it keeps the information of all items in the store, such as the rest area and shoe rack that are marked by red rectangles in Figure 6.6(b). However, the whole map is not readable at all. Because of the poor view range of the Kinect scan; it cannot provide enough information for SLAM to make a good pose estimation. Therefore, the items are laid out in wrong positions in the built map.

The fused map is shown in Figure 6.7. We can see this map has a high consistency (well-registered) with the LIDAR-layer-map, and contains all useful information from the environment. Compared to Figure 6.6(b), we find the use of LIDAR pose reduces the distortion of the map and preserves the useful information. Apparently, the robot will perform much better using the fused map because it will know more precisely its position and the structure of the environment while navigating. And we also mark the shoe rack and rest area with red rectangles in Figure 6.7, as we can see the shoe rack in the map is perfectly shown against the wall as it is, and keep enough profile information for robot navigation.

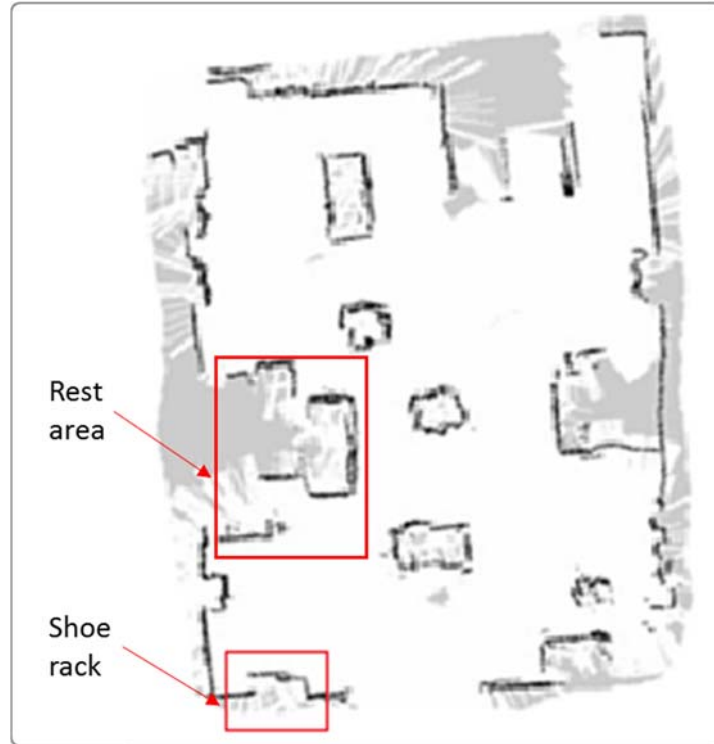


Figure 6.7. Fused map by multiple-layer mapping algorithm with both LIDAR and Kinect scans.

6.3 Navigational Results

To demonstrate the performance of our robot navigating in a retail environment, we also conducted an experiment in our mock store. We used the map that is built above, which is shown in Figure 6.7. In this section, we first take a look at how to generate a global path by our inventory based path planner algorithm, and then we will test the robot to navigate in the store by the generated path.

6.3.1 Global Path Generating

We loaded the map of the store, which is shown in figure 6.7. The map was converted into a costmap, the obstacles in the map were inflated by the inscribed radius of the robot. The result is given as following.

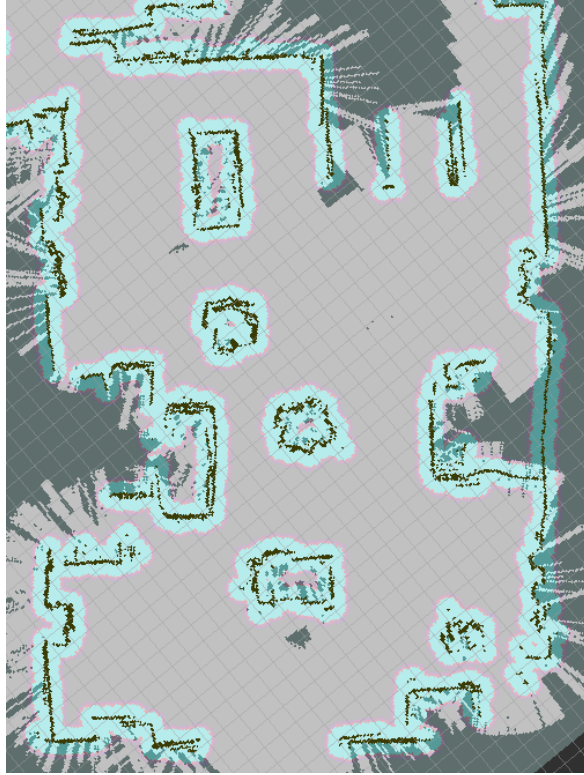


Figure 6.8. Inflated costmap of the experimental mockup store

Then we used the inflated costmap as the input to our global path planner. We used two sets of parameters, $l=0.45$ and $l = 0.8$, which controlled the distance of the global path to the obstacles. The generated global paths are illustrated in Figure 6.9.

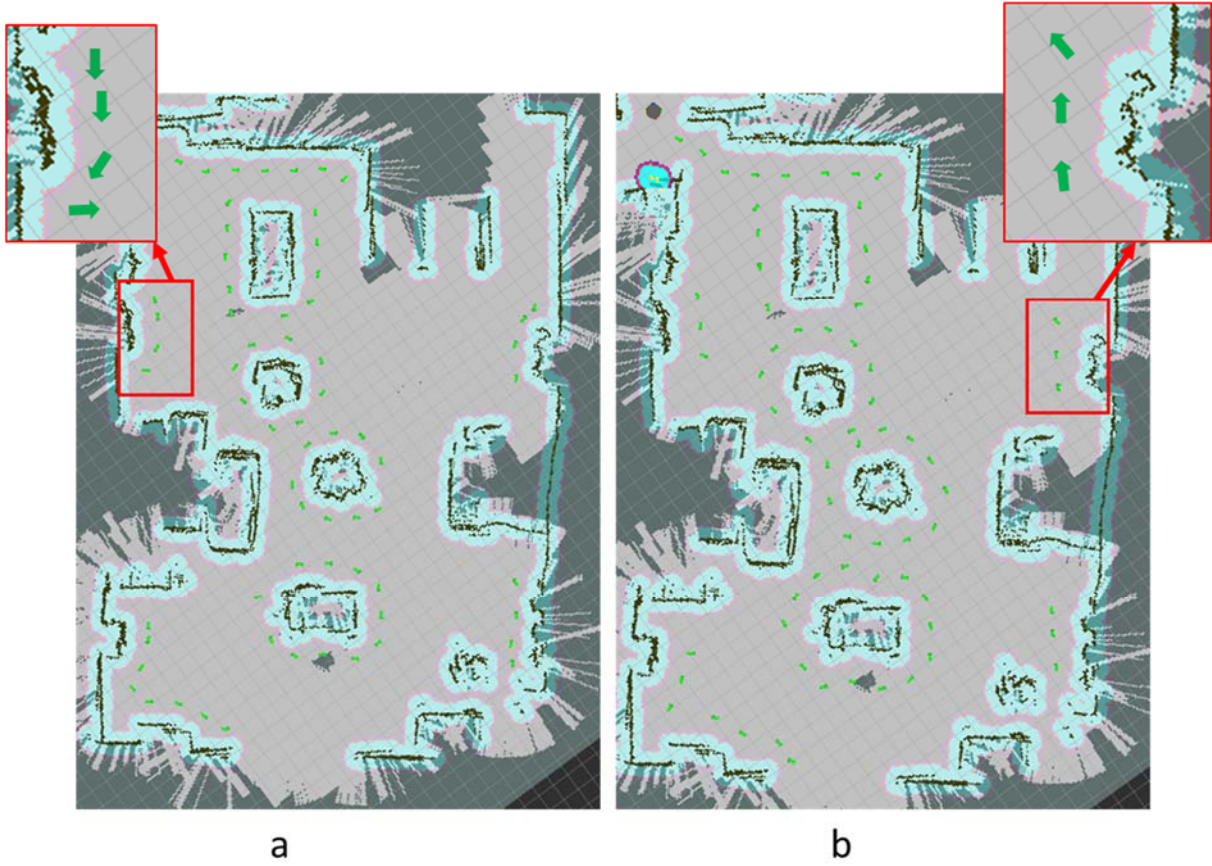


Figure 6.9(a) the global path generated with $l=0.45$, and (b) the global path generated with $l=0.8$. Each generated path is consisted of a group of discrete arrows, which are marked as green in the figure.

We can tell in Figure 6.9, there are total 66 points generated as the global path for $l=0.45$ and 67 points for $l=0.8$. Hereafter, the global path generated under the parameter $l=0.45$ is called the closer path, and the global path generated under the parameter $l=0.8$ is called the farther path. The arrow of each goal point faces to the next point, this will help to smooth the navigation. Both the paths guide the robot to navigate in the merchandise areas; this can greatly improve the navigating performance while not losing any inventory accuracy.

6.3.2 Robot Navigates in the Store

With the map created and the global path generated, the robot is ready for navigating in the store. We navigated the robot by the two global paths, closer path and farther path, which were generated previously. The initial pose of the robot was given. Repeated experiments had demonstrated that the robot can navigate the store reliably while scanning tags in each type of fixture – metal shelves, stacked goods, and metallic hangers. The navigational path resulting from a typical run of each path is shown in Figure. 6.10.

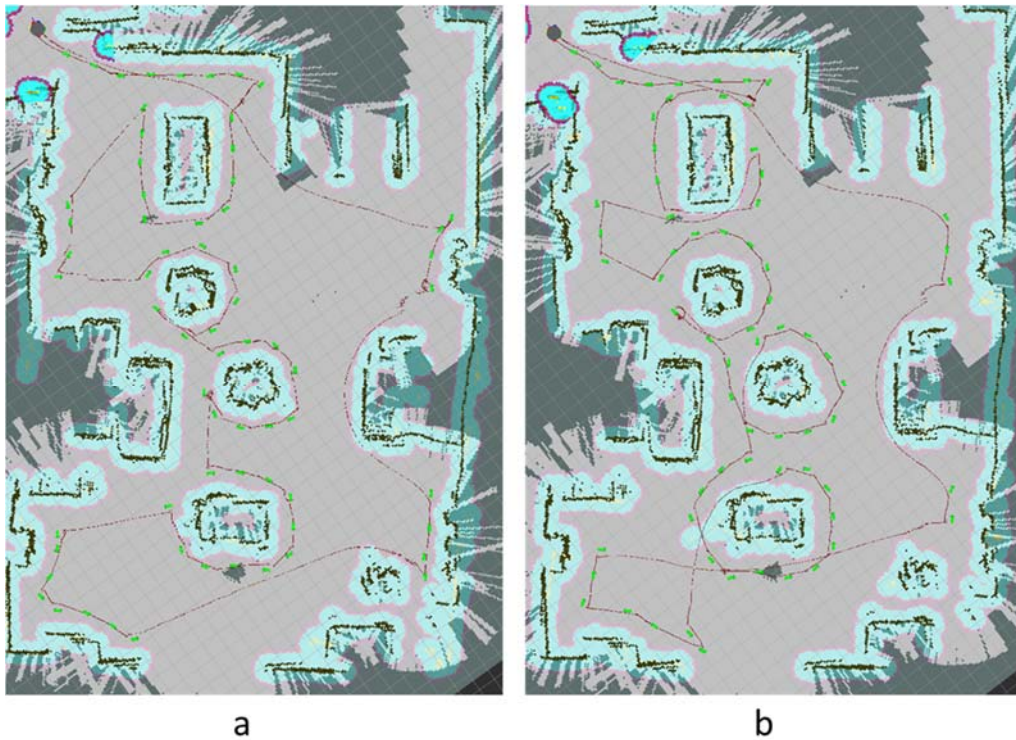


Figure 6.10(a) Robot inventory navigation result of the closer path, (b) Robot inventory navigation result of the farther path

In the above figure, the green arrows in the maps are the global path goals that are autonomously generated by the robot, and the brown curves are records of actual routes. From the routes we can see the robot cycles around the merchandise zones to scan the RFID tags. During the navigation the maximum speed of the robot was limited to 0.8 meter per second. The average

navigation time for the farther path was about 27 minutes; in contrast average navigation time for the closer path was about 30 minutes.

Only depending on the navigational time, we cannot tell which path is a better one, because the ultimate goal of our robot is not navigating in a retail environment but to do inventory. We will compare the performance of them by inventory accuracy, which will be introduced later.

6.4 RFID Inventory Results

In order to compare robot inventory accuracy, we conducted a manual inventory in our mockup store. The RFID reader for our manual inventory experiment was the Zebra RFD8500 handheld, which is a piece of powerful and widely deployed RFID equipment in retail. The manual inventory approach is waving handheld near merchandise. The Zebra RFD8500 handheld and manual inventory approach are illustrated as follows.



Figure 6.11(a) Zebra RFD8500 handheld, (b) waving handheld in front of merchandise to do manual inventory.

During the manual inventory experiment, we set the RFID power to the maximum of the handheld. We repeated the manual experiment five times to get an average accuracy of the result, which is shown in table 6.1.

While the robot was navigating in the store, it collected the RFID tag responses. We used the same map as the navigational experiment and set the reader power to the maximum that is 170 power level. We performed an inventory on our mockup store five times for each global path, to get the mean inventory and mean inventory accuracy shown in the following table.

Table 6.1. Inventory accuracy

Zone	Merchandises type	Ground truth	Closer path		Farther path		Manual	
			Mean inventory result	Mean inventory accuracy	Mean inventory result	Mean inventory accuracy	Mean inventory result	Mean inventory accuracy
Jean shelf	Jeans	370	252	68.11%	240.6	65.03%	187.8	50.76%
Cloth rack	T-shirts, Dress tops, Pants	175	172.8	98.74%	170.2	97.26%	158.4	90.51%
Shoes rack	Shoes	31	31	100.00%	30.8	99.35%	26.8	86.45%
Dresses rack	Dresses, Shoes	23	23	100.00%	23	100.00%	22.6	98.26%
Island rack	Jeans, shoes, T-shirts	45	44	97.78%	44.2	98.22%	37.6	83.56%
Lingerie table	Bras, Panties	30	30	100.00%	30	100.00%	29.8	99.33%
Total		674	552.8	81.78%	538.8	79.70%	463	68.49%

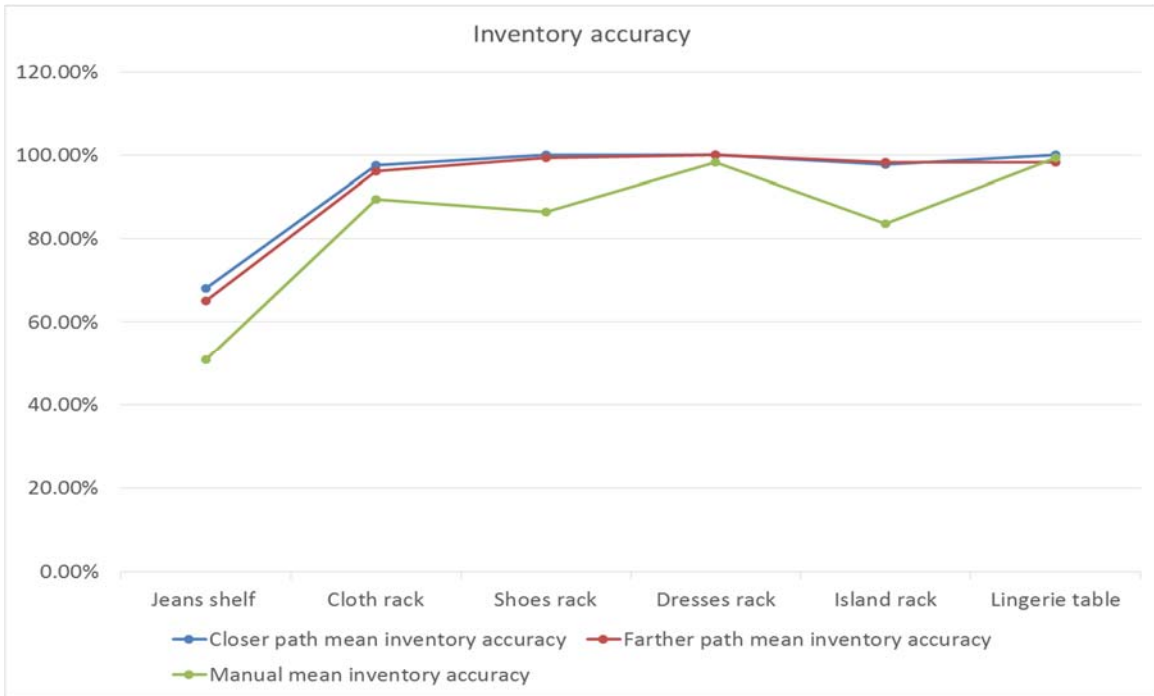


Figure 6.12. Inventory accuracy chart

From Table 6.1, we can see the overall average inventory accuracy for the closer path is 81.78%, except the jean shelf, all other zones are close to 100%. The overall average inventory accuracy for the farther path is 79.70%. In contrast, the manual inventory accuracy is 68.49%. We can tell that the robot inventory is better than the manual one. Because the robot can exactly cover all the target spaces to do a full RFID scanning, while, the manual inventory may leave some places not fully scanned. From figure 6.12 and table 6.1, we also can tell that the performance of the closer path is a little bit better than the farther path.

This experiment also shows that the structure of the merchandise zone can greatly influence the inventory accuracy. From Table 6.1, we see the jean shelf, which is a five layer metal shelf, has the worst inventory results. This shelf is at a height of 76 inches (1.93 m) compared with the mounting height of the upper antennas of 33 inches (0.83 m), which significantly degrades the tag reading. The phenomenon is illustrated in the following figure. This effect also occurs with manual

inventory, because the metal shelf will reflect a lot of the signal waves. Even so, the robotic inventory accuracy is quite competitive with manual inventory.

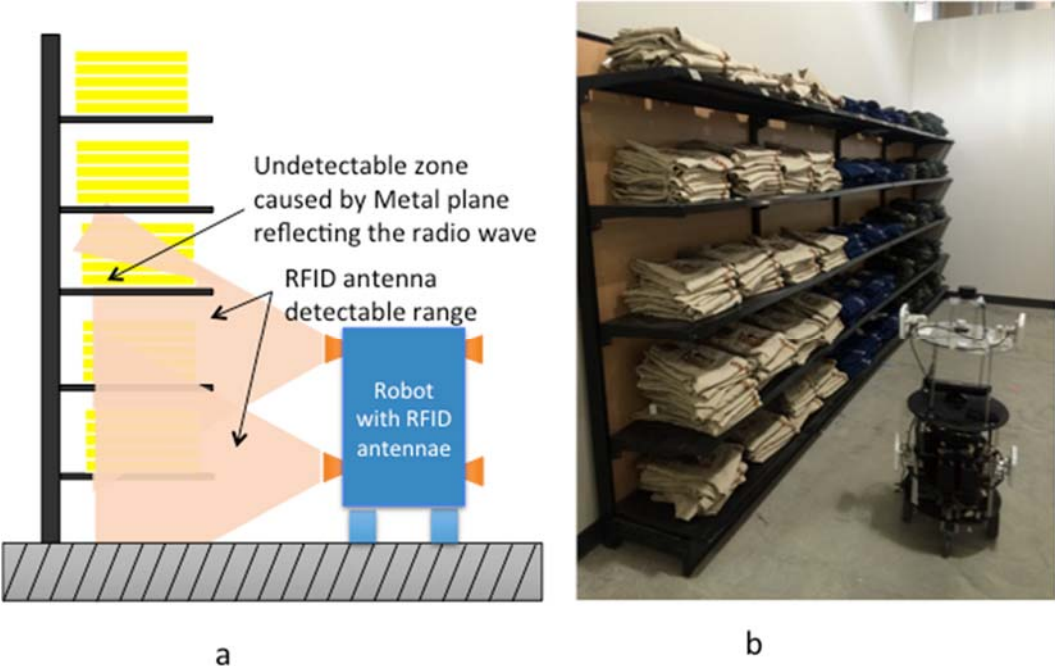


Figure 6.13(a) Illustrates the robot cannot detect the high level RFID tags of the jean shelf, (b) In the photo the robot navigates and collects RFID tags response in front of the jean shelf.

From the above figure, we can tell the robot cannot detect the RFID tags that are placed on a higher level of the jean shelf and those that touch the metal plane in the lower level, and this phenomenon is proved by the experiment data. The following table shows the level inventory accuracy of the jean shelf. More detailed analysis shows that most missed tags are placed in the top level of the metal shelf. We can see in Table 6.2 in the two lower levels the robot gets more than 90% inventory accuracy; in contrast, in the top level the robot only gets 20% inventory accuracy. In the lower level, we find out the RFID tags that are exactly laid on the metal plane are prone to fail detection by the robot, because the metal will shield radio waves.

The similar phenomenon also happened with manual inventory, it gets better inventory accuracy in lower levels. Compared to robot experiment results, the manual one has around 55%

in level 1 to level 4, because when we waved a handheld those levels were mostly covered. In the manual inventory, a human being is not as patient as a robot to cover every space with even and maximum scanning effort; this is the reason why manual inventory accuracy in level 2 to level 4 is lower than the robot. In level 1, however, a human being can elevate the handheld to get better inventory accuracy than the robot. In level 0, the manual experiment got similar inventory accuracy, for the position of level 0 also is too high for a human to cover as the robot does.

Table 6.2. The jean shelf level inventory accuracy

Jean shelf level inventory accuracy							
Jean shelf level	Ground truth	Closer path		Farther path		Manual	
		Mean inventory result	Mean inventory accuracy	Mean inventory result	Mean inventory accuracy	Mean inventory result	Mean inventory accuracy
level 0	59	12	20.34%	11.8	20.00%	10.2	17.29%
level 1	90	42.8	47.56%	41	45.56%	50	55.56%
level 2	77	57.6	74.81%	52.4	68.05%	42.6	55.32%
level 3	84	82.2	97.86%	78	92.86%	53.6	63.81%
level 4	60	57.4	95.67%	57.4	95.67%	31.4	52.33%
total	370	252	68.11%	240.6	65.03%	187.8	50.76%

Notes: Level 4 is closest to the ground, Level 0 is farthest from the ground

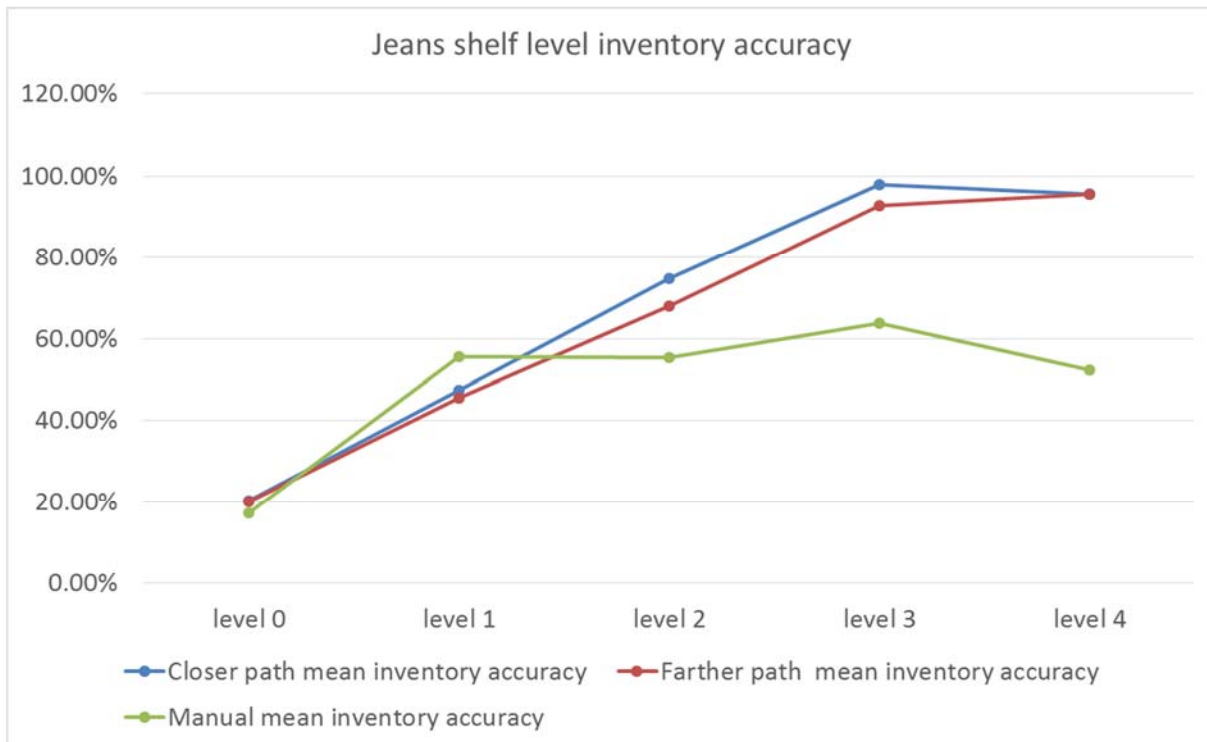


Figure 6.14. Jean shelf level inventory accuracy chart

An intuitive solution to improve the inventory of high vertical structure is to mount a RFID antenna in a higher position on the robot.

6.5 RFID Tag Localization Results

Our last experiment series demonstrate the performance of the RFID tags localization. We still conducted our experiments in the mock store. We will test our two RFID tags localization algorithms separately, fixed power tag localization and variable power tag localization algorithm, and compare the results by localization accuracy. The localization accuracy is presented by the error between the estimated location and the ground truth position of each RFID tag. The ground truth position is manually measured by locating each item in a given map. The ground truth and the estimated position are both measured in a 2-dimensional Cartesian coordinate system referenced to the given map.

We conducted 4 experiments for fixed power tag localization. In these 4 experiments during the observation collecting phase, the RFID reader was operated at 80, 110, 140 and 170 power levels separately. Then we can compare the localization results for each experiment. We use the map same as the navigational experiment and only use the closer path, because when conducting our experiment under the lower power level, we need the robot to move as close to the merchandise as possible, so that we can collect more RFID tag responses. The average observation collecting time is in the same level of RFID inventory, which is about 30 minutes.

During the experiment of the variable power tag localization algorithm, we used the goal points of the closer path as the measurement points in the observation collecting phase. The measurement points are illustrated in the following figure.

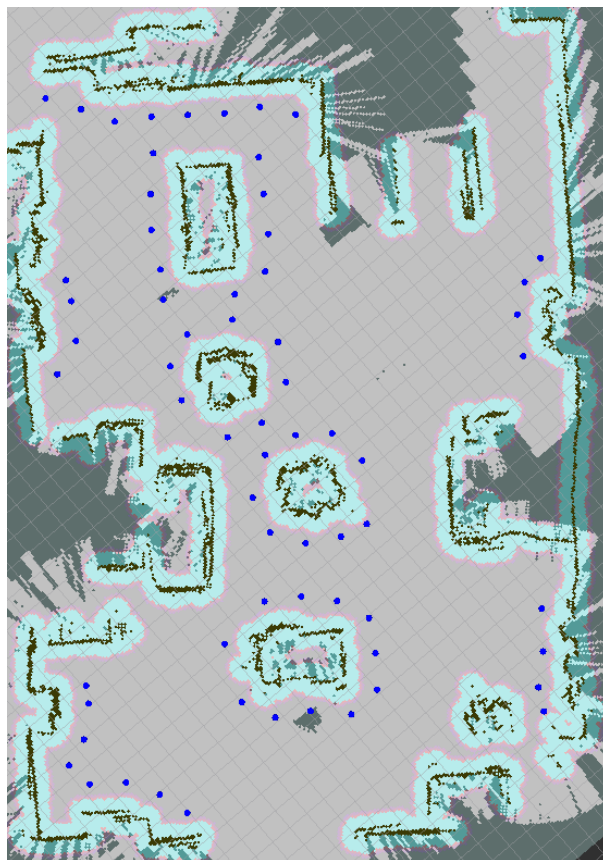


Figure 6.15. The blue points represent the measurement points of variable power tag localization

The robot will stop at each measurement point and shift ± 90 , ± 45 and 0 degrees to collect RFID tag responses under 80, 110, 140 and 170 power level. Base on those RFID tag responses we can generate the measurement vector for the variable power tag localization algorithm. The average observation collecting time is about 6 hours. The results are given in the following table.

Table 6.3. Tag localization results

Zone	fixed power tag localization reader power 170			fixed power tag localization reader power 140			fixed power tag localization reader power 110			fixed power tag localization reader power 80			variable power tag localization		
	rate	average error (meter)	standard deviation	rate	average error (meter)	standard deviation	rate	average error (meter)	standard deviation	rate	average error (meter)	standard deviation	rate	average error (meter)	standard deviation
Jeans shelf	35.14%	0.831	0.527	23.24%	0.886	0.539	10.27%	0.632	0.275	5.68%	0.462	0.17	38.11%	0.486	0.267
Cloth rack	100.00%	1.005	0.439	98.53%	0.707	0.335	73.53%	0.702	0.266	38.24%	0.596	0.254	100.00%	0.496	0.261
Shoes rack	100.00%	1.634	1.248	85.71%	0.886	0.472	39.29%	0.918	0.543	14.29%	0.69	0.197	100.00%	0.483	0.184
Dresses rack	100.00%	0.692	0.234	95.65%	0.685	0.55	86.96%	0.536	0.263	69.57%	0.497	0.267	100.00%	0.401	0.179
Island rack	92.00%	0.992	0.459	64.00%	0.943	0.543	36.00%	0.773	0.277	18.00%	0.909	0.23	94.00%	0.507	0.261
Lingerie table	100.00%	1.783	0.543	100.00%	1.371	0.508	100.00%	1.093	0.398	34.62%	1.038	0.516	100.00%	0.543	0.199

notes: in our localization experiment the ground truth number may be slight different from our inventory experiment one

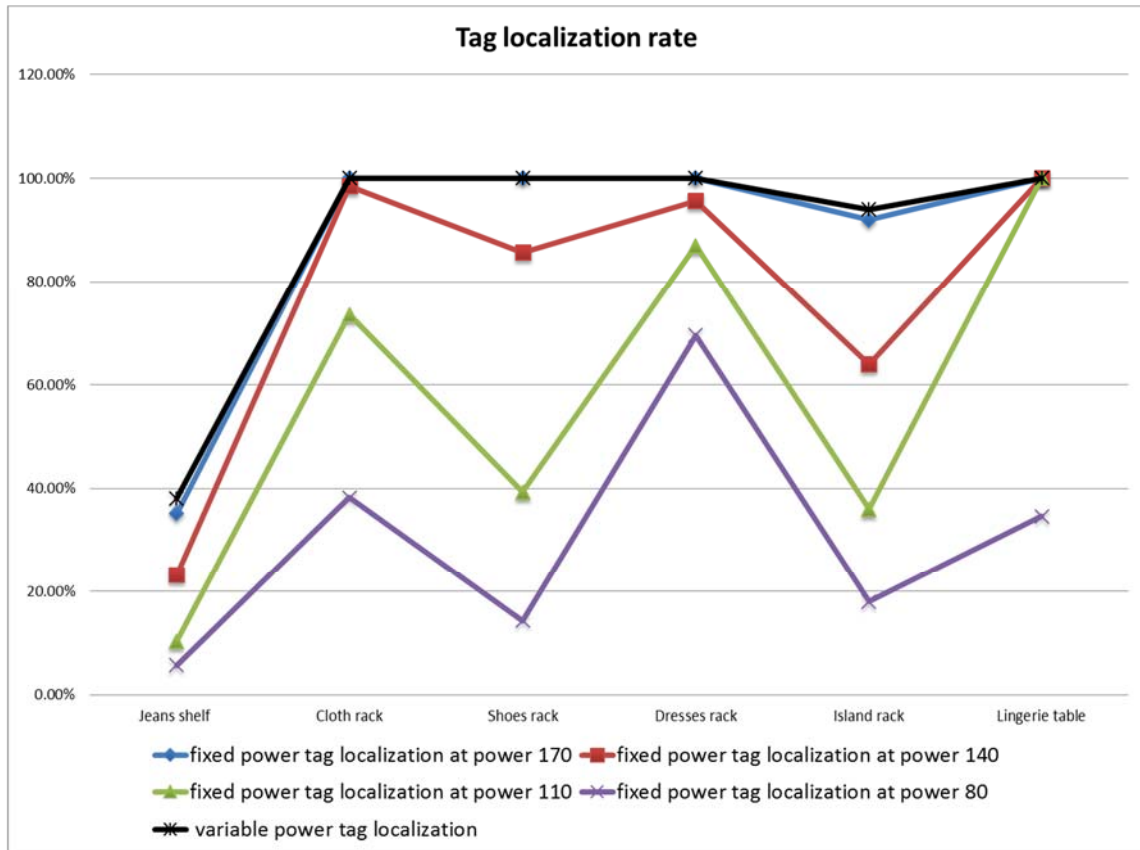


Figure 6.16. Tag localization rate by different algorithms

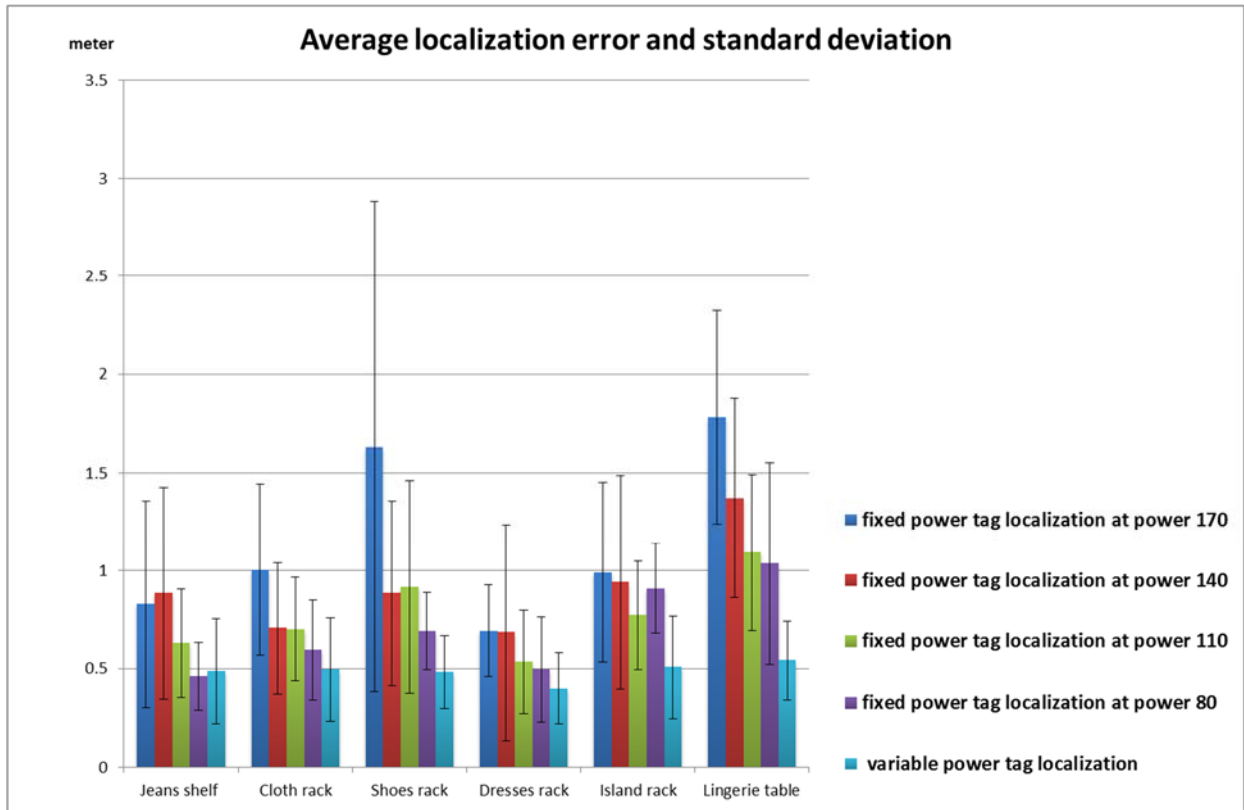


Figure 6.17. Tag localization average error and standard deviation by different algorithms

There are two important criteria to evaluate RFID tags localization methods:

1. The tag localization rate, the percentage of localized tags to all existing tags.
2. The localization error, which can be measured by average error and standard deviation in a zone.

From Table 3, Figure 6.16 and Figure 6.17, the localization rate of the fixed power tag localization algorithm is degraded greatly as the RFID reader power is decreased. However, the localization error is decreased as we reduce the RFID reader power. We can tell that the variable power tag localization algorithm provides better performance of localization. It got the same localization rate with the fixed power localization algorithm at the strongest RFID reader power, and smaller localization error and standard deviation than all fixed power localization algorithms. This is because the variable power tag localization algorithm utilizes vector-based

measurement with responses under the variable power level. The vector-based measurement can provide longer detectable range while providing higher localization resolution.

The experiment results tell that the structure of the merchandise zones can greatly influence the tag localization rate, too. The jean shelf got very low tag localization rate, because only the items in the lower levels, which is level 3 and 4, got enough observations to be localized. The experiments also tell us that the variable power tag localization algorithm provides very reliable localization accuracy. In all merchandise zones the average error is about 0.5 meter with about 0.2 meter standard deviation. The localization accuracy provides great practical application value; the 0.5 meter accuracy is enough for localization of an item in the retail environment, the following figure shows how the localization results help to locate items in our mockup store.

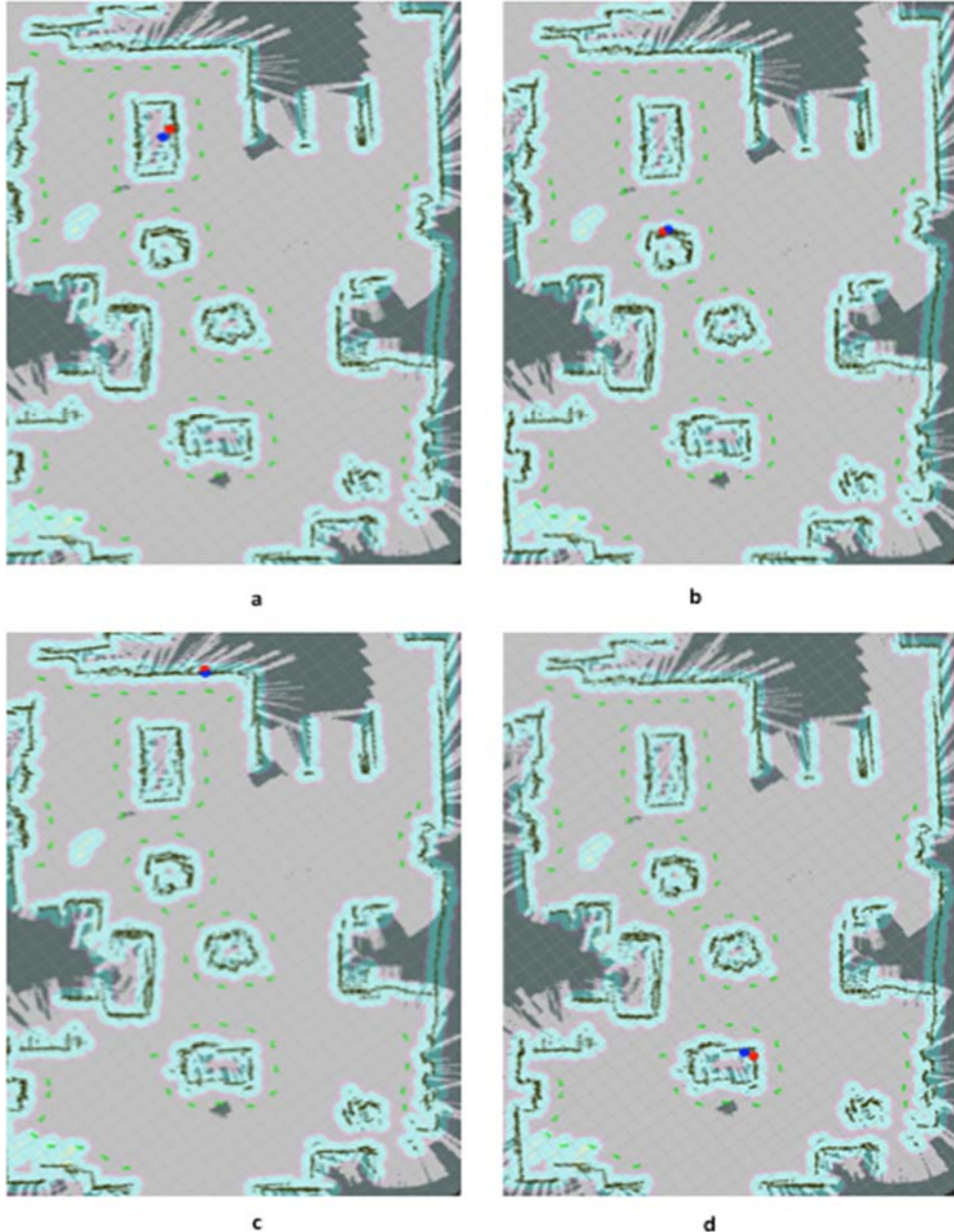


Figure 6.18. Examples of localized items in our mock store. The blue points are the ground truth positions of items, and red points are the positions estimated by the variable power tag localization algorithm. (a). an undergarment is localized on the lingerie table, (b) a T-shirt is localized in one of cloth racks, (c) a pair of jeans is localized in the jeans shelf, and (d) a pair of jeans is localized in the island rack.

The above figure tells that combined with a given map, the localization results of the variable power tag localization algorithm can help to find items in the sales floor or warehouse. For

example in (d) of Figure 6.18 the localized result tells that the pair of jeans are placed on the right corner of the island rack. And this localization result can be achieved on mass numbers of RFID tags in a real store environment.

But the fixed power tag localization algorithm is more efficient than the variable one. It only takes about 30 minutes to collect and evaluate all the tag localizations, but variable power tag localization algorithm needs 6 hours to do so. The results tell us that these two algorithms are fitted for two different scenarios. The fixed power tag localization algorithm can be used to quickly estimate the location of merchandise. For example to locate many items in a giant warehouse, the results of the fixed power tag localization algorithm are enough to locate those items. If more precise localization results are needed, the variable power tag localization algorithm is the better choice. For example, if we need find misplaced items on the sales floor, we could apply a robot to do the localization task during off time, and the precise localization results can be generated. The precise localization results could help locate a misplaced item to the exact position to reduce the effort of labor to find it and relocate it.

Chapter 7 Conclusion and Future Work

This dissertation presents a novel application of a mobile robot that performs retail inventory autonomously and provides merchandise positions by localizing the attached RFID tags. The robot builds a map by using onboard sensors in an enclosed, indoor environment. The robot then generates a global path that can navigate the robot to collect RFID tag responses with its onboard RFID reader and antennae. We showed that the robot can efficiently do the inventory in the retail sales floor, which consists of different types of merchandise and structures. The inventory accuracy is better than that of manual inventory. We also developed two RFID tag localization algorithms for the robot. The fixed power tag localization algorithm can be used to quickly estimate the location of RFID tags, but it only provides limited localization accuracy. The variable power tag localization algorithm needs the robot to do some extra effort to collect observations of RFID tags; however, it provides very precise localization accuracy that could guide a human being to find any item in a retail environment.

7.1 Summary

We have conducted several experiments in our mock apparel store to evaluate the performance of our robot to operate in retail environments. The results of inventory experiments show that the robot can efficiently navigate in the obstacle-rich environment. The inventory accuracy is highly competitive with the manual one. The results also show that if we can improve our robot structure by raising the vertical height of RFID antennae to enable them to cover all merchandise, the robot could yield even better inventory accuracy. Therefore, the

results indicate that deploying a robot or a group of robots to perform retail inventory represents a promising alternative to deploying a large number of stationary RFID readers or having personnel scan objects manually. Our approach greatly reduces the manual labor required in retail stores for inventory. In addition, it is significantly less expensive than equipping every possible location with an RFID reader antenna [62].

We also compared the performance of two RFID tag localization algorithms, the fixed power and the variable power tag localization algorithms. The results show that the fixed power tag localization algorithm takes about 30 minutes to estimate all the location of items in our mock store; in contrast, the variable power method needs about 6 hours to do so. The variable power tag localization algorithm provides much better localization accuracy, the average error is about 0.5 meter. Under the same localization rate, the fixed power tag localization algorithm could only provide meter level localization accuracy. Therefore, two algorithms are fitted for two different application scenarios: the fix power tag localization algorithm can be used to quickly estimate the location of merchandise, and the variable power tag localization algorithm could be deployed for a scenario that needs more precise localization results.

Combining the results of inventory and tag localization experiments, we arrive at the conclusion that our robot could be deployed in real retail environments. It could greatly reduce the effort of inventory, and help to improve the efficiency of the retail inventory management by generating merchandise locations.

7.2 Future Work

One of the main difficulties for deploying the robot to giant retail environments is that a single robot cannot complete the inventory task in a short time. Currently the robot needs about 30 minutes to complete the inventory in 204 square meters of our mockup store. The

supercenters of Wal-Mart could size up to 260,000 square feet [61], which is about 24,154 square meters, and may have multiple floors. Apparently, a single robot is not enough to complete the inventory task in an acceptable time period. Our future work will focus on developing a group of robots to do inventory in the same space, this will greatly reduce the time for doing inventory in giant retail environments such as the sales floor of a Wal-Mart supercenter.

Another future work will focus on improving the tag localization algorithm, that does not depend on the robot platform to collect the RFID observations. For example, we can collect the tag responses by a handheld, and based on this kind of observation, the tag localization algorithm could estimate the locations of target items with RFID tags.

References

- [1] Hardgrave, Bill, "Try It, You'll Like It! - The RFID Lab's annual state-of-adoption report of U.S. retailers," RFID Journal, Aug. 25, 2015.
- [2] Dellaert, Frank, et al. "Monte carlo localization for mobile robots." Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on. Vol. 2. IEEE, 1999.
- [3] Fung, W. K., et al. "Development of a hospital service robot for transporting task." Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on. Vol. 1. IEEE, 2003.
- [4] BBC News. Robot cleaner hits the shops. May 2003. url: <http://news.bbc.co.uk/2/hi/technology/3031219.stm>.
- [5] William Woodall and Michael Carrol. "Moe: the Autonomous Lawnmower". In: ROSCON, St. Paul Minnesota. May 2012.
- [6] Hine, Roger, et al. "The wave glider: A wave-powered autonomous marine vehicle." Proc. of MTS/IEEE OCEANS. 2009.
- [7] Ehrenberg, Isaac, Christian Floerkemeier, and Sanjay Sarma. "Inventory management with an RFID-equipped mobile robot." 2007 IEEE International Conference on Automation Science and Engineering. IEEE, 2007.

- [8] Schairer, Timo, et al. "Integrated scenario for machine-aided inventory using ambient sensors." RFID Systems and Technologies (RFID SysTech), 2008 4th European Workshop on. VDE, 2008.
- [9] Melià-Seguí, Joan, et al. "Enhancing the shopping experience through RFID in an actual retail store." Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. ACM, 2013.
- [10] Mead, Ross, et al. "An architecture for rehabilitation task practice in socially assistive human-robot interaction." 19th International Symposium in Robot and Human Interactive Communication. IEEE, 2010.
- [11] Marder-Eppstein, Eitan, et al. "The Office Marathon. Robust Navigation in an Indoor Office Environment ". In: International Conference on Robotics and Automation. (2010).
- [12] Stäubli. PUMA. 2014. url: <http://www.staubli.com/en/robotics/>.
- [13] Davids, Angela. "Urban search and rescue robots: from tragedy to technology." IEEE Intelligent Systems 17.2 (2002): 81-83.
- [14] Mao, Guoqiang, ed. Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking: Monitoring and Surveillance Techniques for Target Tracking. IGI Global, 2009.
- [15] Latombe, Jean-Claude. Robot motion planning. Vol. 124. Springer Science & Business Media, 2012.
- [16] Moravec, Hans P. "Sensor fusion in certainty grids for mobile robots." AI magazine 9.2 (1988): 61.

- [17] Hu, Huosheng, and Michael Brady. "A Bayesian approach to real-time obstacle avoidance for a mobile robot." *Autonomous Robots* 1.1 (1994): 69-92.
- [18] Ó'Dúnlaing, Colm, and Chee K. Yap. "A "retraction" method for planning the motion of a disc." *Journal of Algorithms* 6.1 (1985): 104-111.
- [19] Borenstein, Johann, and Yoram Koren. "The vector field histogram-fast obstacle avoidance for mobile robots." *IEEE Transactions on Robotics and Automation* 7.3 (1991): 278-288.
- [20] Borenstein, Johann, and Yoram Koren. "Real-time obstacle avoidance for fast mobile robots in cluttered environments." *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on.* IEEE, 1990.
- [21] Thrun, D. Fox W. Burgard S., D. Fox, and W. Burgard. "The dynamic window approach to collision avoidance." *IEEE Transactions on Robotics and Automation* 4 (1997): 1.
- [22] Choset, Howie. "Coverage for robotics—A survey of recent results." *Annals of mathematics and artificial intelligence* 31.1-4 (2001): 113-126.
- [23] Colegrave, J., and A. Branch. "A case study of autonomous household vacuum cleaner." *AIAA/NASA CIRFFSS* 107 (1994).
- [24] Huang, Y., Z. Cao, and E. Hall. "Region filling operations for mobile robot using computer graphics." *Robotics and Automation. Proceedings. 1986 IEEE International Conference on.* Vol. 3. IEEE, 1986.
- [25] Land, Stephanie, and Howie Choset. "Coverage path planning for landmine location." *Third International Symposium on Technology and the Mine Problem.* 1998.
- [26] Hofner, Christian, and Günther Schmidt. "Path planning and guidance techniques for an autonomous mobile cleaning robot." *Intelligent Robots and Systems' 94.'Advanced Robotic*

Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on. Vol. 1. IEEE, 1994.

[27] Elfes, Alberto. "Sonar-based real-world mapping and navigation." IEEE Journal on Robotics and Automation 3.3 (1987): 249-265.

[28] Lumelsky, Vladimir J., Snehasis Mukhopadhyay, and Sun Kang. "Dynamic path planning in sensor-based terrain acquisition." IEEE Transactions on Robotics and Automation 6.4 (1990): 462-472.

[29] Bessho, Masahiro, et al. "Location systems for ubiquitous computing." J. IEICE 92.4 (2009): 249-255.

[30] Pahlavan, Kaveh, Xinrong Li, and Juha-Pekka Makela. "Indoor geolocation science and technology." IEEE Communications Magazine 40.2 (2002): 112-118.

[31] Fang, Bertrand T. "Simple solutions for hyperbolic and related position fixes." IEEE transactions on aerospace and electronic systems 26.5 (1990): 748-753.

[32] Drane, Christopher, Malcolm Macnaughtan, and Craig Scott. "Positioning GSM telephones." IEEE Communications Magazine 36.4 (1998): 46-54.

[33] Nikitin, Pavel V., et al. "Phase based spatial identification of UHF RFID tags." 2010 IEEE International Conference on RFID (IEEE RFID 2010). IEEE, 2010.

[34] Zhou, Junyang, KM-K. Chu, and JK-Y. Ng. "Providing location services within a radio cellular network using ellipse propagation model." 19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers). Vol. 1. IEEE, 2005.

- [35] Teuber, Andreas, Bernd Eissfeller, and Thomas Pany. "A two-stage fuzzy logic approach for wireless LAN indoor positioning." Proc. IEEE/ION Position Location Navigat. Symp. Vol. 4. 2006.
- [36] Van Veen, Barry D., and Kevin M. Buckley. "Beamforming: A versatile approach to spatial filtering." IEEE assp magazine 5.2 (1988): 4-24.
- [37] Ottersten, B., et al. "Exact and large sample ML techniques for parameter estimation and detection in array processing, Ch. 4 in" Radar Array Processing" Simon Haykin." (1993).
- [38] Wang, Xuyu, et al. "CA 2 T: Cooperative Antenna Arrays Technique for Pinpoint Indoor Localization." Procedia Computer Science 34 (2014): 392-399.
- [39] Wang, Xuyu, et al. "DeepFi: Deep learning for indoor fingerprinting using channel state information." 2015 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2015.
- [40] Wang, Xuyu, et al. "CSI-based Fingerprinting for Indoor Localization: A Deep Learning Approach." (2016).
- [41] Wang, Xuyu, Lingjun Gao, and Shiwen Mao. "PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach." 2015 IEEE Global Communications Conference (GLOBECOM). IEEE, 2015.
- [42] Chawla, Vipul, and Dong Sam Ha. "An overview of passive RFID." IEEE Communications Magazine 45.9 (2007): 11-17.
- [43] Stockman, Harry. "Communication by means of reflected power." Proceedings of the IRE 36.10 (1948): 1196-1204.
- [44] Roberti, Mark. "The history of RFID technology." RFID Journal 16.01 (2005).

- [45] https://en.wikipedia.org/wiki/Electromagnetic_induction
- [46] Raab, Frederick H. "High-efficiency linear amplification by dynamic load modulation." Microwave Symposium Digest, 2003 IEEE MTT-S International. Vol. 3. IEEE, 2003.
- [47] http://digilander.libero.it/windflow/eng/Usage/RFIDSensor/rfid_sensor.htm
- [48] Dobkin, Daniel M. The rf in RFID: uhf RFID in practice. Newnes, 2012.
- [49] Dissanayake, MWM Gamini, et al. "A solution to the simultaneous localization and map building (SLAM) problem." IEEE Transactions on Robotics and Automation 17.3 (2001): 229-241.
- [50] Khoshelham, Kourosh, and Sander Oude Elberink. "Accuracy and resolution of kinect depth data for indoor mapping applications." Sensors 12, no. 2 (2012): 1437-1454.
- [51] Murphy, Kevin, and Stuart Russell. "Rao-Blackwellised particle filtering for dynamic Bayesian networks." Sequential Monte Carlo methods in practice. Springer New York, 2001. 499-515.
- [52] Thrun, Sebastian, et al. "Robust Monte Carlo localization for mobile robots." Artificial intelligence 128.1 (2001): 99-141.
- [53] Carpenter, James, Peter Clifford, and Paul Fearnhead. "Improved particle filter for nonlinear problems." IEE Proceedings-Radar, Sonar and Navigation 146, no. 1 (1999): 2-7.
- [54] <http://www.robotshop.com/en/rplidar-360-laser-scanner.html>
- [55] Canny, John. "A computational approach to edge detection." IEEE Transactions on pattern analysis and machine intelligence 6 (1986): 679-698.

- [56] Thrun, D. Fox W. Burgard S., D. Fox, and W. Burgard. "The dynamic window approach to collision avoidance." IEEE Transactions on Robotics and Automation 4 (1997): 1.
- [57] Raman, Ananth, Nicole DeHoratius, and Zeynep Ton. "Execution: The missing link in retail operations." California Management Review 43.3 (2001): 136-152.
- [58] Hariharan, Sharethram. "Improving Dynamic Decision-Making Through RFID: A Partially Observable Markov Decision Process (POMDP) for RFID-Enhanced Warehouse Search Operations." (2006).
- [59] Friis, Harald T. "A note on a simple transmission formula." Proceedings of the IRE 34.5 (1946): 254-256.
- [60] <http://wiki.ros.org/gmapping>
- [61] <http://247wallst.com/retail/2014/03/22/walmart-now-has-six-types-of-stores/>
- [62] Zhang, Jian, et al. "Mobile robot for retail inventory using RFID." 2016 IEEE International Conference on Industrial Technology (ICIT). IEEE, 2016.