國立交通大學

資訊學院　資訊學程

碩 士 論 文

以 RFID 及 WiFi 定位技術設計自我導航機器人系統

Designing Self-Guiding Robot with Assistance of RFID tags and WiFi Localization System

研 究 生：徐敏修

指導教授：曾煜棋　教授

中 華 民 國 九 十 九 年 一 月

以 RFID 及 WiFi 定位技術設計自我導航機器人系統
Designing Self-Guiding Robot with Assistance of RFID tags and WiFi Localization System

研 究 生：徐敏修　　Student：Ming-Hsiu Hsu

指導教授：曾煜棋　　Advisor：Yu-Chee Tseng

國 立 交 通 大 學

資訊學院　資訊學程

碩 士 論 文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

January 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年一月

以 RFID 及 WiFi 定位技術設計自我導航機器人系統

學生：徐敏修　　　　　　　　　　　　指導教授：曾煜棋老師

國立交通大學　資訊學院　資訊學程碩士班

摘　要

我們想建立一套應用在家庭或辦公空間的室內移動機器人系統。在家庭或辦公空間，物品位置常會有變動的情形發生。所以，機器人的自我導航的能力對機器人是否能順利到達指定位置成為了關鍵的因素。在整合了無線通訊，室內定位和 RFID 技術後，我們可以建立一套低成本的自我導航機器人系統。我們提出了一個架構，可以在公分級到公尺級的精準度中達成此目標。首先，機器人可以下載目前區域的室內地圖，再透過 WiFi 定位方式來決定公尺等級的初步位置，進一步，經由下載的室內地圖，機器人可以透過我們設計的搜尋的方式，嚐試讀取預先設置的 RFID 地標。當機器人實際讀取到 RFID 地標時，就可以將自己的位置設定到公分等級的精準度並提供良好的服務了。為了驗證我們提出方法的可行性，我們以 iRobot 平台開發了一個雛型系統。由實驗的結果，這個方法在未來家庭和辦公環境中會有很好的應用機會。

關鍵字：室內定位，定位方法，遍佈式運算，RFID，機器人

Designing Self-Guiding Robot with Assistance of RFID tags and WiFi Localization System

student：Ming-Hsiu Hsu　　Advisors：Prof.Yu-Chee Tseng

Degree Program of Computer Science
National Chiao Tung University

ABSTRACT

We consider building an indoor low-cost mobile robot that can be used in home or office applications. Due to the complication and dynamic changing nature of a home/office environment, it is essential for such a robot to be self-guiding in the sense that it is able to determine its current location as well as navigate to locations which it is commanded to. By integrating with wireless communication, indoor localization, and RFID technologies, it is possible to build a self-guiding robot at low cost. We propose an architecture to achieve this goal at centimeter-to-meter level accuracy. A robot can even roam into an area which is new to it. We achieve this goal by allowing a robot to download the indoor map in its vicinity. With WiFi localization, it determines its rough location at meter-level accuracy. From the downloaded map, it then tries to track any RFID tag in the indoor map, on which events it actually localizes itself at centimeter-level accuracy. To verify our results, we demonstrate a prototyping system based on iRobot. Our results have potentially important implications on future digital home/office.

Keywords: indoor positioning, localization, pervasive computing, RFID, robot

誌 謝

首先，誠摯的感謝曾煜棋教授兩年多來對我的細心指導和鼓勵。曾教授對於研究的熱情，經由每天規律而專注的生活來實踐，達到了很好的成果。這個身教，讓我感受很多。曾教授所帶領的實驗室也提供了良好的研究環境，讓我能夠獲得充分的硬體環境的支援，以順利的完成此篇論文研究並且順利取得碩士學位。

也由衷的感謝葉倫武學長在論文的研究與寫作上，持續的指導我一步步的建構這篇研究論文到完成。實驗室的家良、育萱、志偉和阿寬在我論文寫作和實驗的過程中也都給了我很多的協助和鼓勵，在此謝謝 HSCC 實驗室的大家。 而我們一起創造的許多美好回憶和建立的感情，也是我這次研究所生活中的最大收獲。

最後，感謝我的家人以及我身邊的朋友，在這段時間對我的支持和體諒。你們的支持，讓我無後顧之憂的完成學業。

Contents

List of Figures

Chapter 1

Introduction

Indoor mobile robots have lots of potential application in daily life, such as home care, room cleaning, object finding, and emergency supporting. Most mobile robot services would require a robot to determine its current location. While this issue has been intensively studied in factory environments, it is more challenging for homes/offices, where the environment is not so strictly controlled and the interior planning is subject to change at any time. The purpose of this study is to design a suitable indoor navigation mechanism for mobile robots to meet this need.

To precisely navigate for a robot, a localization method is needed. GPS dominates many outdoor localization applications. However, for indoor localization, a globally usable solution is still missing. In Section 2, we review some existing indoor localization solutions. In this work, we propose a framework to design "self-guiding" indoor robot. Our goal is to utilize assistance of current technologies, such as RFID, wireless networks, and indoor localization systems to achieve this goal. We list some design guidelines of the framework:

- Plug-and-play: The robot does not require to have any pre-knowledge about the environment before entering the environment. After entering the environment, its needs not to explore the entire environment before it can get started.

- Self-guiding: The robot needs not to know its initial location in this environment. This means that the robot might be moved manually to any location at any time and it can still recalibrate itself.

- Multi-level localization accuracy: Depending of different situations, the system can provide different levels of accuracy, perhaps at different costs.
- Ease of installation: The extra infrastructures required should be as minimal as possible and, if possible, we may rely on existing infrastructure.
- Low cost: we are not looking for complicated systems. It should be inexpensive and affordable.
- Scalability: Our system should be scalable without adding much overheads on the robot itself.

In order to achieve above feature, we propose a self-guiding robot with assistance of RFID and WiFi localization systems. To make the robot "self-guiding", we attach a WiFi card and a RFID reader on the robot and deploy numerous RFID tags on the ground. A location server based on WiFi fingerprinting method can help the robot to position itself at meter-level precision. Besides, the pre-installed tag map can provide the robot to position itself at centimeter-level precision. Through the self-guiding mechanism, the robot can support centimeter-level moving requirement and need not equip the expensive components. Then, the self-guiding robot can provide indoor navigation services automatically.

The remainder of this work is organized as follows. In Section 2, we briefly describe the RFID and WiFi localization system and survey several related works about indoor robot navigation. Section 3 presents our system design of self-guiding robot system. Section 4 further describes the system implementation details and several evaluations. Section 5 demonstrates our system by two case studies. Finally, Section 6 concludes this work.

Chapter 2

Preliminaries

2.1 Introduction to RFID

RFID (Radio Frequency Identification) is a general term that is used to describe a system that transmits the identity of an object via wireless. Recent year, RFID has been used in many situations in our daily life, such as tracking goods, animal identification, passport, etc.

A typical RFID system is composed of three components: tags, readers and the host computer. 1) Tags: It is a tiny radio device. The tag is composed of a simple silicon microchip attached to a small flat antenna and mounted on a substrate. The tags can be attached on an object, such as smart cards, passports, etc. 2) Readers: It can send or receive RF data to or from the tag via antenna. The action can be achieved without direct line of sight. 3) Host computer: The data acquired by the readers is then passed to a host computer. The host computer can run several RFID software to process the obtained information.

The RFID tags can be divided into two types: active and passive tags. 1) Active tags: The active tags have internal power source. With internal power source, it can transmit information longer than passive tags. The active tags usually operate at 455 MHz, 2.45 GHz, or 5.8 GHz, and their communication range is about 20 to 100 meters. 2) Passive tags: The passive tags have no power source and no transmitter. The passive tags usually operate at 125 kHz, 13.56 MHz, or 915

MHz. These tags are more cheaper than active tags and require no maintenance. But, the communication range of passive tags are more shorter than active tags (about 3 to 5 centimeter or 10 meter).

2.2 Introduction to Indoor Localization

Localization is a critical issues for indoor environment. GPS dominates the outdoor localization, but, for indoor localization, a globally usable solution is still missing. Many indoor localization technologies have been proposed, such as infrared-based [1], ultrasonic-based [2], and RF-based [3] systems. Generally, localization models can be classified as AoA-based [4], ToA-based [5], TDoA-based [6], and fingerprint-based [3][7][8].

Recent year, WiFi becomes the most common wireless transmission media in our daily life. Hence, in this work, we are interested in fingerprint-based localization systems, such as RADAR [3], by WiFi transmission media. This method does not rely on calculating signal fading in an environment. Instead, it relies on a training phase to collect the radio signal strength (RSS) patterns at a set of training locations from pre-deployed beacons in a sensing field into a database (called radio map). These beacons can be existing infrastructures, such as IEEE 802.11 access points, GSM base stations, or other RF-based networks. Then, during the positioning phase, an object to be localized can collect its current RSS pattern and compare it against the radio map established in the training phase to identify its possible location.

2.3 Indoor Robot Navigation

Mobile robot can provide several services in indoor environment and most of these services should require robot's current location. Hence, how to precisely locate and navigate the mobile robot is very important topic. In the past, indoor robot navigation systems can be divided into three categories: (1) laser guiding system

[9], (2) magnetic tape guiding system [10], and (3) vision-based system [11]. In [9], the robot can receive at least three laser guiding signal to determine its current location. Laser and magnetic tape guiding systems are more accurate but allow only limited routes. Vision-based systems are more flexible, but they require high computing cost and expensive equipments.

Recently, RFID techniques have also been applied to robot navigation [12][13][14]. In [12][14], an array of short range passive RFID tags are deployed on the entire field. Each tag represents a unique location. On scanning these tags, the robot can calculate its location. Grid deployment is discussed in [12], while triangular deployment is discussed in [14]. As can be seen, both [12] and [14] will need a large number of RFID tags. The work [13] uses a rotatable RFID reader to guide a robot to a stationary target, which has an active tag as a location transponder. An AoA-like model is adopted to guide the robot. This solution is more costly and it is mainly for one single target. Our work uses passive tags and it takes advantage of existing WiFi networks for meter-level positioning.

Chapter 3

Design of a Self-Guiding Robot

3.1 System Architecture of the Self-Guiding Robot

Fig. 3.1 shows the system architecture of our self-guiding robot system. There are four main components: WiFi network, location server, RFID landmarks, and robot.

- WiFi network: The network contains a number of access points, which periodically transmit beacons. By collecting signal strengths of these beacons, a device can estimate its current location in a meter-level accuracy.

- Location server: The location server runs a fingerprinting algorithm, which provides meter-level accuracy, for positioning purpose. It also keeps some maps and those locations on the maps where RFID tags are deployed.

- RFID landmarks: RFID tags serve as landmarks, each identifying a unique location for centimeter-level localization. These tags are divided into groups and deployed on several key points of the floor. On scanning a tag, the robot get its current location on the map in a centimeter-level accuracy.

- Robot: The self-guiding robot is a moving service platform which has a WiFi interface and a RFID reader. It decides its location in two phases. Through collected RSS patterns, it can determine its rough location on a

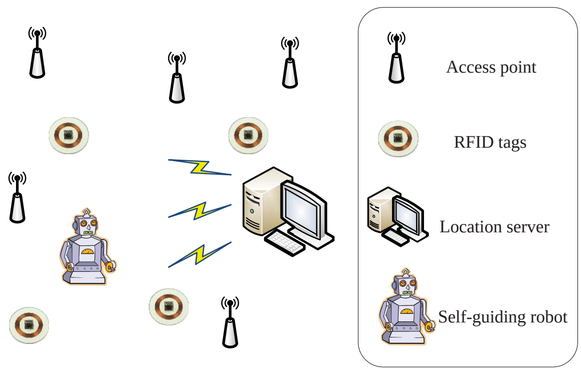


Figure 3.1: The system architecture of self-guiding robot system.

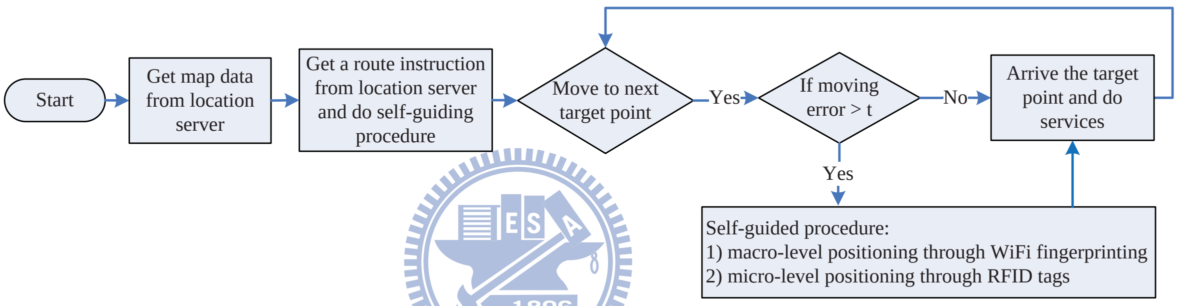


Figure 3.2: The system flow of self-guiding robot.

map. According to the map, it can further search for RFID tags. Once a tag is found, it knows its precise location. Besides, the robot also can equip other sensors, such as camera, compass, thermometer, etc., to collect more information while providing services.

3.2 System Flow of the Self-Guiding Robot

Fig. 3.2 shows the system flow of self-guiding robot. Below, we describe the detail steps of system flow.

1. Initially, robot does not know any pre-knowledge about this area. Hence, the robot will download the map data, including topology information and

RFID tags map, from location server via wireless media. If the robot is the first time into this area, it will do the self-guiding procedure to locate itself.

2. After getting the map data, the robot will get a *route instruction*, which contains a sequence of target points.

3. The robot tries to move to the next target point in the route instruction. If there is no next entry in the route instruction, the robot will get the next route instruction to do services.

4. Due to continuous moving during a long time or external influences, such bumping into obstacle, picking up by human being, etc, the robot will lose its precise location. Hence, before moving to the target point, the robot checks the accumulated moving error and relocates itself. If the accumulated moving error greater than a predefined threshold t, the robot will do the self-guiding procedure to identify its accuracy location. Otherwise, the robot tries to move to the expected target point.

5. The self-guiding procedure contains two level localization steps. The two levels are macro-level positioning (through WiFi fingerprinting) and micro-level positioning (through RFID tags).

   (a) In the first level, the robot collects current WiFi signal strength and transmits to location server via wireless media. Then, the fingerprinting algorithm can be executed by the location server to estimate the location of the robot at meter-level precision. Since RF signal is inherently unstable, the fingerprinting algorithm only can locate robot in a meter-level precision.

   (b) In the second level, the robot tries to scan the RFID tags to locate its precise location in a centimeter-level precision. Through the map data and current estimated location, the robot can move to the RFID tags directly. However, due to the error of the first level, the robot may

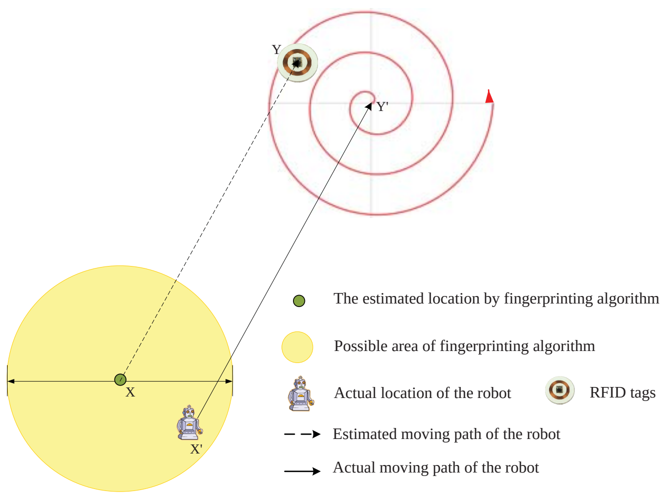


Figure 3.3: An example of self-guiding procedure.

not find the RFID tags directly. Then, the robot searches the field in a spiral manner to find the RFID tags. If any RFID tags is scanned, its current location can be decided at the centimeter-level precision.

6. The robot arrives the target point and does some predefined services, such as collecting current RSSI, temperature, light degree, etc. When the robot finishes the predefined services, it will go to step 3 to move to the next target point.

Note that in the self-guiding procedure, when the accumulated moving error greater than E, where E is the maximum error of fingerprinting localization, the robot does the two level self-guiding procedure. Otherwise, the robot only does the second level self-guiding procedure. Fig. 3.3 shows an example of self-guiding procedure. At first, the robot gets its estimated location X by fingerprinting algorithm. However, due to the error of fingerprinting algorithm, the actual location of the robot is X'. The robot tries to move form X to Y (the dotted line) to scan the RFID tags. But, in actually, the robot moves from X' to Y' (the straight line). When the robot move to Y', the robot does not find the RFID tags. The robot starts to search the RFID tags in a spiral manner. Finally, through the spiral search, the

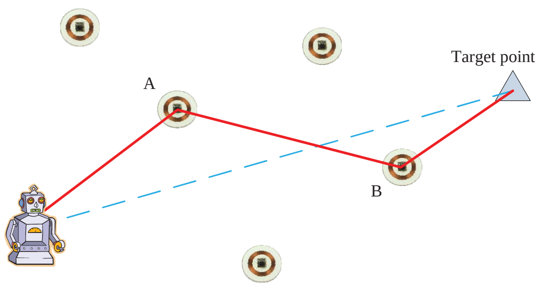


Figure 3.4: An example of long distance moving.

robot finds the RFID tags in location Y.

Note that in step 3, the robot can not move to the target point in a straight line directly when the distance between robot and target point is greater than a threshold l. This is because that the accumulated moving error may greater than E while moving to the target point on the way. In this case, the robot can insert some intermediate points (RFID tags) into the routing path. The robot moves to the intermediate points (RFID tags) one by one until arriving the target point. The problem of inserting the appropriate intermediate points can adopt geographic routing algorithm in wireless ad hoc network [15][16]. The target point and the tolerant accumulated moving error are corresponding to the destination node and communication range in wireless ad hoc network, respectively. Fig. 3.4 shows an example of long distance moving. The robot moves to the intermediate points A and B then moves to the target point.

3.3 Discussions of Spiral Search

We adopt fingerprinting algorithm in the first level of self-guiding procedure to get estimated location. Since the RF signal is inherently unstable, the estimated location may have meter-level error. According to the result of fingerprinting algorithm, the possible locations are presented in Gaussian distribution. Hence, we adopt the Archimedean spiral [17], which is successive turnings of the spiral in a constant separation distance, in our system. The robot can search the RFID

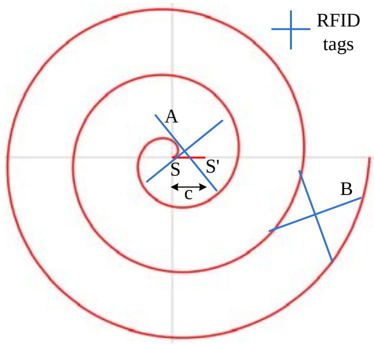


Figure 3.5: An example of spiral search.

landmark in a spiral manner to identify its location.

We deploy the RFID landmarks as the cross shape. In some cases, the robot can not scan the RFID tags quickly. Fig. 3.5 shows an example of Archimedean spiral and the starting point is point S. From starting point S, the robot will miss to scan the nearest RFID tags group A. This is because the starting angle of Archimedean spiral is too small. Hence, we add a constant straight line c to solve this problem. As shown in Fig. 3.5, the robot moves from starting point S' to S then starts to run the Archimedean spiral.

Through the spiral search, the robot can locate itself in a expected time. We assume that the robot moves in a constant velocity. Then, the upper bound of total moving distance, denoted as $spiral(\theta)$, while doing spiral search method can be decided, where θ is the total moving degree of spiral angle.

Property 1. *In the robot spiral search method, given the maximum error of fingerprinting algorithm E, constant separation distance d, constant c, radius r, and polar angle θ. The upper bound of total moving distance is*

$$spiral(\theta) \leq \frac{1}{2}d[\frac{E}{d}\sqrt{1+\frac{E^2}{d^2}} + \ln(\frac{E}{d}+\sqrt{1+\frac{E^2}{d^2}})] + c.$$

Proof. 1. In polar coordinate (r, θ), the Archimedean spiral can be described by the equation $r = d\theta$.

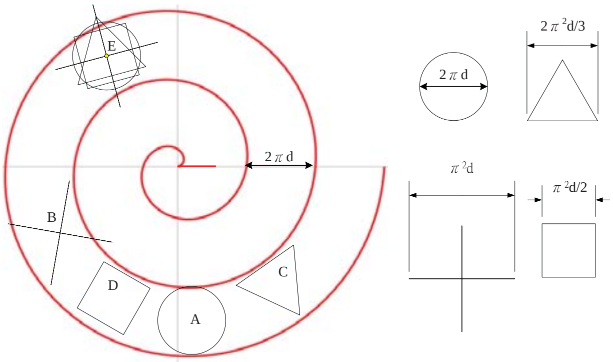


Figure 3.6: Different kinds of RFID tags groups.

2. Given the θ value, the arc length of Archimedean spiral can be denoted as follows:

$$spiral(\theta) = \frac{1}{2}d[\theta\sqrt{1+\theta^2} + \ln(\theta + \sqrt{1+\theta^2})].$$

3. As we know, the maximum error of fingerprinting algorithm is E. Hence, we can bound the Archimedean spiral radius to E, i.e., $r = d\theta \leq E$. According to above equations and the initial straight line constant c, we can get the upper bound of total moving distance as follows:

$$spiral(\theta) \leq \frac{1}{2}d[\frac{E}{d}\sqrt{1+\frac{E^2}{d^2}} + \ln(\frac{E}{d} + \sqrt{1+\frac{E^2}{d^2}})] + c.$$

□

3.4 Deployment Cost of RFID Tags

Our landmark groups are composed of several RFID tags. Different shape of landmarks need different number of RFID tags. It means different shape needs different cost. Fig. 3.6 shows an example of different kinds of RFID tags groups. Here, we set the separation distance of Archimedean spiral to d. We set the diameter of circle to $2\pi d$, which is the same as the constant separation distance of Archimedean spiral.

With the same deployment cost (i.e., circumference of the shape), we can get the width of cross shape, equilateral triangle, and square are $\pi^2 d$, $2\pi^2 d/3$, and $\pi^2 d/2$, respectively. As shown in Fig. 3.6, equilateral triangle (i.e., shape C) and square (i.e., shape D) are not intersection with Archimedean spiral in some cases. Although circle (i.e., shape A) can intersect with Archimedean spiral anywhere, the width of circle is less than cross shape. The width of cross shape is larger than others and it can intersect with spiral anywhere. By the cross shape RFID tags groups, we can guarantee that robot can scan RFID tags group certainly and may scan earlier than other shapes. As shown in Fig. 3.6, the point E shows an example of different shapes in the same location. Robot can scan the cross shape with spiral path earlier than others.

Chapter 4

Implementation and Evaluation

4.1 Hardware and Software Components

Fig. 4.1(a) and Fig. 4.1(b) show the components of the self-guiding robot. We adopt iRobot Create [18] as the mobile platform and attach a RFID reader, an electronic compass, and a notebook with WiFi interface. The iRobot Create is a programmable moving platform which is more cheaper and flexible than others. The iRobot Create includes a cargo bay which contains a 25 pin port that can be used for digital and analog input and output. We connect a RS232 multi-port controller on the cargo bay. The RFID reader is short range reader which operates in 13.56 MHz and the acceptable reading range is limited within 5 cm. The iRobot, RFID reader, and electronic compass are connected by the RS232 multi-port controller. Through the multi-port controller, the robot can get the location and orientation information from RFID reader and electronic compass, respectively. Fig. 4.1(c) shows the shape of RFID landmarks, which are deployed into a cross shape with 30×30 cm. Each tag contains its unique ID, X and Y offset values.

Fig. 4.2 shows our functional blocks of implementation. The implementation architecture can be divided into three parts: *Device Layer*, *Robot Controlling Layer*, and *Location Server*. The Device Controller Layer can get data from different devices respectively. The Robot Controlling Layer contains the Self-guiding

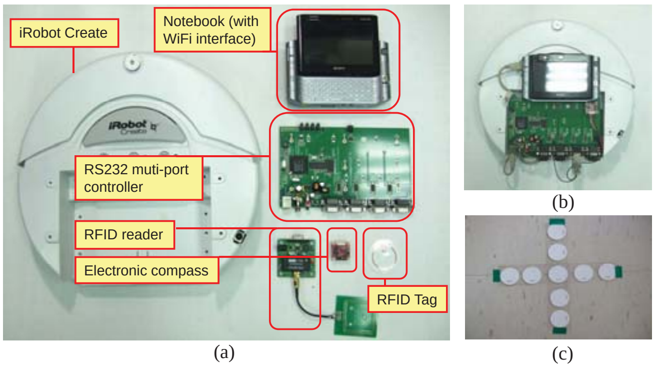


Figure 4.1: (a) Hardware components of self-guiding robot. (b) The self-guiding robot. (c) RFID tags groups.

Procedure and Moving Control block which can get location information and map data from Location Server. The Location Server contains a database which stores a set of RSSI readings for fingerprinting algorithm.

4.2 Implementation Issues

There are several issues during designing and implementing our system. We discuss these issues and solutions in the following parts.

1. Communication delay: The mechanism to get data from the robot is polling operation. Initially, we send a request to start the operation. The device will execute all the time and periodically report the current status until we send the stop command to the device. The highest speed to send the request is 19200 bps, as the result, the communication delay is about $500 \sim 600$ milliseconds. The communication delay can not be neglected in this operation due to our accurate requirements. Hence, before running this system, we try to record the delays of every kinds of operations. During running this system, we send the stop command according the recorded values beforehand to relieve the influence of communication delays. Fig. 4.3 shows an example. Here, we send a start command to trigger robot moving. The

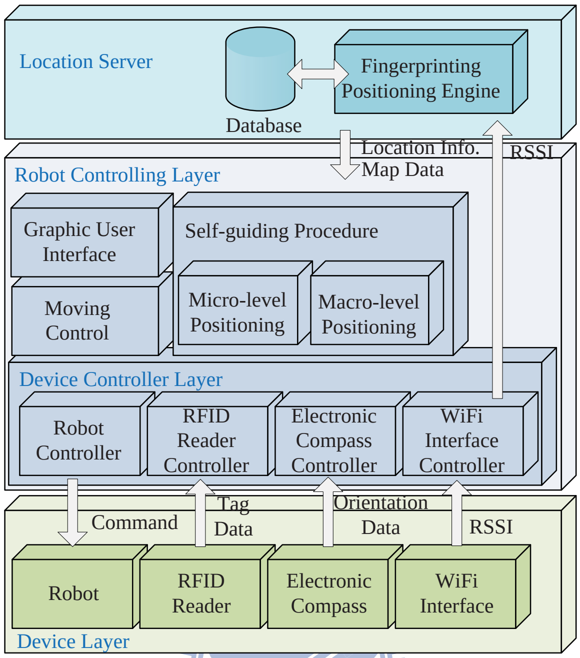


Figure 4.2: The functional blocks of implementation.

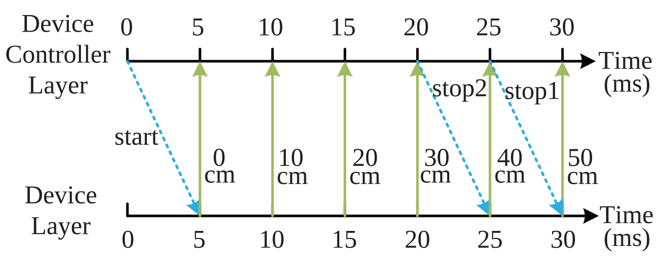


Figure 4.3: An example of communication delay.

communication delay of sending command is 5 ms. The Device Layer periodically reports current distance to Device Controller Layer every 5 ms. Now, we want to stop the robot at 40 cm. We send the stop command (dash line stop1) after receiving the report containing 40 cm. Due to communication delay, the robot will stop at 50 cm. If we send the stop command (dash line stop2) at 20 ms, the robot will stop at 40 cm.

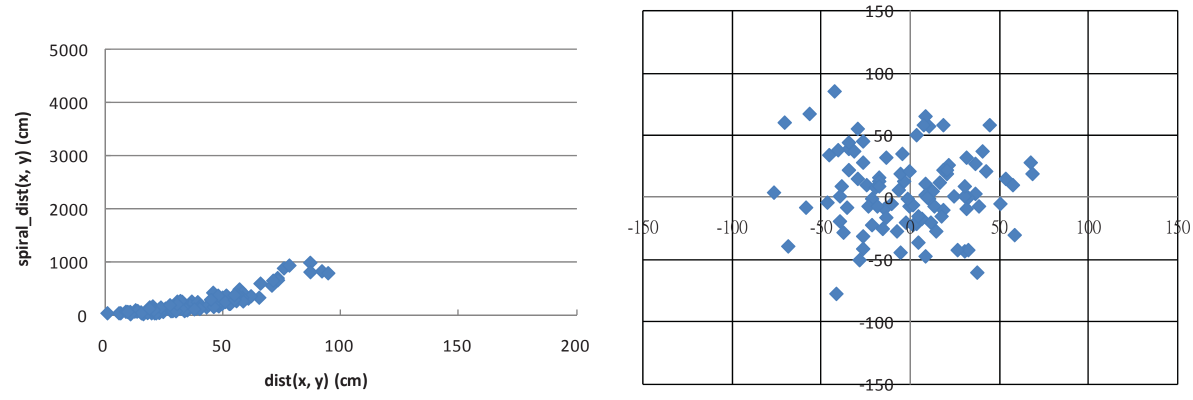
2. Heavy loading: The driving wheels mechanism of iRobot is coaxial gear. So, if we attach too heavy objects on the robot, the robot will tilt several degrees. It will cause the moving error after long distance moving. For example, we attach a small notebook about 1 kg on the robot. It will cause 1 meter accumulated moving error after moving 20 meters. Hence, we attach light weight notebook and controller modules on the robot to relieve this problem.

3. Response time of RFID reader: When robot moves above the RFID tags, the RFID reader on the robot needs to negotiate with the RFID tags to get the information. The negotiation time is direct proportional with the data length in RFID tags. Hence, we control the moving speed of robot and limit the data length in RFID tags to avoid the negotiation fail, which means the robot is across the RFID tags, but do not get any information. We only record the tag type, landmark group ID, offset value, X and Y axis in the RFID tags.

4.3 Evaluations

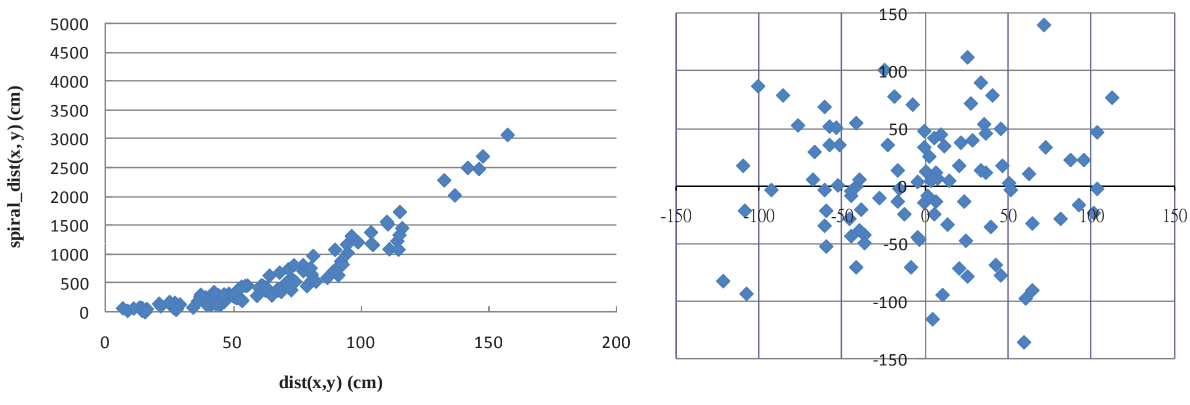
In this section, we evaluate our system by simulations and implementations.

We evaluate the performance of the micro-level positioning under different precision of the macro-level positioning. In macro-level positioning, the possible locations are presented in Gaussian distribution. Hence, we randomly produce the estimated location (x, y) by Gaussian distribution function $f(\mu, \sigma) =$



(a) (b)

Figure 4.4: (a) The simulation result under $(\mu, \sigma) = (0, 30)$. (b) An example of generating points under $(\mu, \sigma) = (0, 30)$.



(a) (b)

Figure 4.5: (a) The simulation result under $(\mu, \sigma) = (0, 50)$. (b) An example of generating points under $(\mu, \sigma) = (0, 50)$.

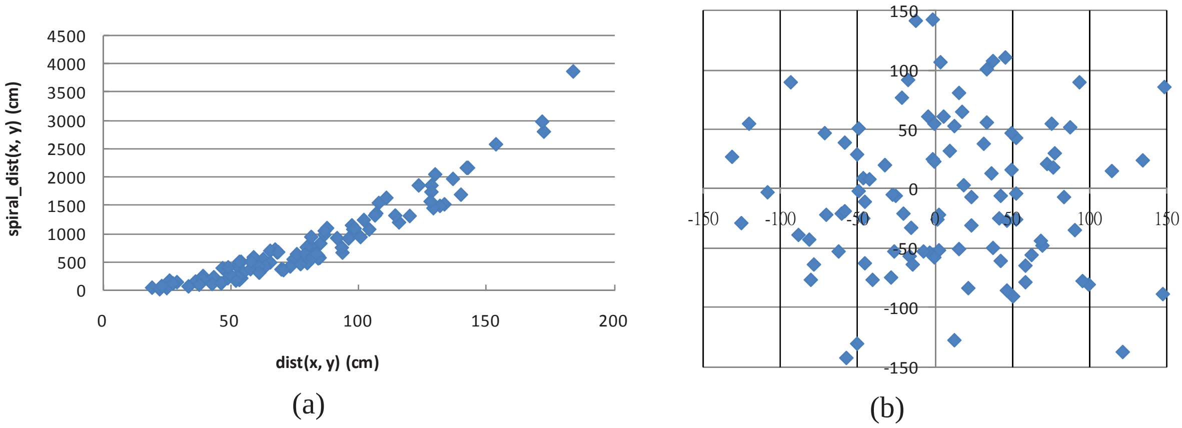


Figure 4.6: (a) The simulation result under $(\mu, \sigma) = (0, 70)$. (b) An example of generating points under $(\mu, \sigma) = (0, 70)$.

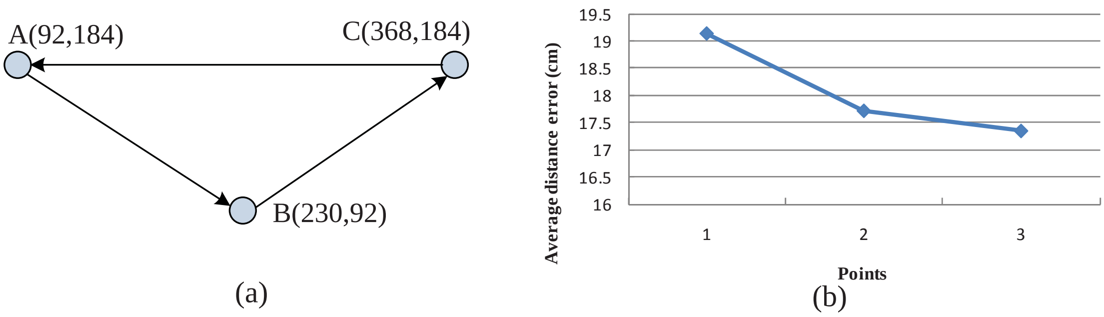


Figure 4.7: (a) An experimental scenario. (b) The experimental result of average distance error.

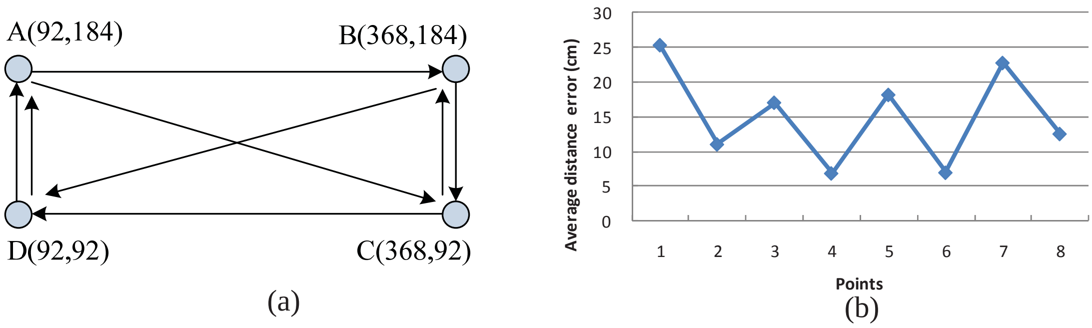


Figure 4.8: (a) An experimental scenario. (b) The experimental result of average distance error.

$exp(\frac{-(i-\mu)^2}{2\sigma^2})$, where $i = x$ or y. We set (μ, σ) to $(0, 30)$, $(0, 50)$, and $(0, 70)$, and randomly produce 100 points within $0 < i < 150$ cm. Fig. 4.4(b), Fig. 4.5(b), and Fig. 4.6(b) show these generating points. Then, the robot starts from the center to search these points in a spiral manner. The $dist(x, y)$ and $spiral\_dist(x, y)$ denote the distance of center to a point in a straight and spiral manner, respectively. Fig. 4.4(a), Fig. 4.5(a), and Fig. 4.6(a) show the result of searching these points under different (μ, σ) values. The results show that the smaller σ values have shorter searching distances. The smaller σ values imply more accurate macro-level positioning. Therefore, the more accurate macro-level positioning results can make our micro-level positioning more quick to find the target.

Next, we evaluate the performance of self-guiding procedure. As shown in Fig. 4.7(a) and Fig. 4.8(a), we deploy three and four groups of RFID tags, respectively, and the robot moves around these points many times in a predefine sequence. When robot moves to a RFID tags, it will relocate itself by our proposed self-guiding procedure. Fig. 4.7(b) and Fig. 4.8(b) show the results of moving errors. The results show that our proposed self-guiding procedure indeed can guide robot while traversing many target points.

Chapter 5

Case Studies

Our system proposes a robot can guide itself in an indoor environment. It can apply to many kinds of scenarios in our daily life. In the following, we present two case studies for our system.

5.1 Case 1: Guiding System

There are many large exhibitions in our daily life. In the exhibition, people want to know the contents of these exhibits. Also, everyone is interested in different kinds of exhibits in an exhibition. To solve this problem, we apply our system in this scenario. We deploy landmark RFID tags, exhibit RFID tags, access points, a location server and a self-guiding robot in the exhibition. The landmark RFID tags group and access points are used to micro-level positioning and macro-level positioning, respectively. Except the landmark RFID tags group, we also deploy exhibit RFID tags containing the ID of exhibits nearby the exhibits. The location server contains the map information, landmark RFID tags information and the exhibits RFID tags information. The self-guiding robot can guide the users to visit the exhibition and also can dynamically plan the visiting routes according to users' preference exhibits.

5.2 Case 2: Automatically Constructing Today's Radio Map

For indoor localization, a globally usable solution is still missing. Here, we are interested in the fingerprinting-based solution, such as [3][7][8]. The major drawback of the fingerprinting approach is its labor-intensive training process, especially in a large-scale field. Further, since the RF signal is inherently unstable, the radio map collected earlier may deviate significantly from the current one. Manually calibrating radio maps is error-prone. Besides, environment may change, furniture may be moved, and beacons may be reconfigured or upgraded anytime [19]. The radio signal strength may vary at different times. This may result in non-negligible errors when positioning objects. Hence, we adopt the self-guiding robot to collect the signal strength automatically. The robot can move around to automate the training process and, more importantly, to frequently update radio maps to reflect the current RSS patterns. This not only significantly reduces human labors but also improves positioning accuracy.

Chapter 6

Conclusions

In this work, we design an indoor low-cost mobile robot that can be used in home or office applications. The robot can relocate itself by our proposed self-guiding procedure in indoor environment. We implement the whole system on iRobot Create and evaluate it by several experiments. Two case studies present our system can be used in our daily life. These results show that our work are potentially important implications on future digital home/office.

Bibliography

[1] E. Brassart, C. Pegard, and M. Mouaddib, "Localization using infrared beacons," *Robotica*, vol. 18, no. 2, pp. 153–161, 2000.

[2] N. B, Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. of ACM Int'l Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[3] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proc. of IEEE INFOCOM*, 2000.

[4] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using aoa," in *Proc. of IEEE INFOCOM*, 2003.

[5] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a sentient computing system," *IEEE Computer*, vol. 34, no. 8, pp. 50–56, 2001.

[6] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine grained localization in ad-hoc networks of sensors," in *Proc. of ACM Int'l Conference on Mobile Computing and Networking (MobiCom)*, 2001.

[7] S.-P. Kuo and Y.-C. Tseng, "A scrambling method for fingerprint positioning based on temporal diversity and spatial dependency," *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 678–684, 2008.

[8] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Chen, "Multidimensional vector regression for accurate and low-cost location estimation in pervasive comput-

ing," *IEEE Trans. on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1181–1193, 2006.

[9] K. Lingemanna, A. Nuchtera, J. Hertzberga, and H. Surmannb, "High-speed laser localization for mobile robots," *Robotics and Autonomous Systems*, vol. 51, no. 4, pp. 275–296, 2005.

[10] Automated guided vehicle. [Online]. Available: http://en.wikipedia.org/wiki/Automated_Guided_Vehicle

[11] E. M. Lowa, I. R. Manchesterb, and A. V. Savkina, "A biologically inspired method for vision-based docking of wheeled mobile robots," *Robotics and Autonomous Systems*, vol. 55, no. 10, pp. 769–784, 2007.

[12] B.-S. Choi, J.-W. Lee, and J.-J. Lee, "An improved localization system with rfid technology for a mobile robot," in *Proc. of conference of IEEE Industrial Electronics (IECON)*, 2008.

[13] M. Kim and N. Y. Chong, "Rfid-based mobile robot guidance to a stationary target," in *Mechatronics*, 2007.

[14] H. Lim, B. Choi, and J. Lee, "An efficient localization algorithm for mobile robots based on rfid system," in *Proc. of Int'l Joint Conference on SICE-ICASE*, 2006.

[15] P. Enge and P. Misra, "Special issue on global positioning system," *Proc. of the IEEE*, vol. 87, no. 1, pp. 3–15, 1999.

[16] M. Witt and V. Turau, "Bgr: Blind geographic routing for sensor networks," in *In Proceedings of the Third Workshop on Intelligent Solutions in Embedded Systems*, 2005.

[17] Archimedean spiral. [Online]. Available: http://en.wikipedia.org/wiki/Archimedean_spiral

[18] iRobot. [Online]. Available: http://store.irobot.com/corp/index.jsp

[19] S.-P. Kuo, H.-J. Kuo, and Y.-C. Tseng, “The beacon movement detection problem in wireless sensor networks for localization applications,” *IEEE Trans. on Mobile Computing*, to appear.