國 立 交 通 大 學

資訊科學與工程研究所

博 士 論 文

藉由無線感測網路來設計與實做
一套智慧型生活空間

Design and Implementation of a Smart Living Space
by Wireless Sensor Networks

研 究 生：葉倫武

指導教授：曾煜棋　教授

中 華 民 國 九 十 九 年 十 月

藉由無線感測網路來設計與實做一套智慧型生活空間
Design and Implementation of a Smart Living Space by Wireless Sensor Networks

研 究 生：葉倫武　　Student：Lun-Wu Yeh

指導教授：曾煜棋　　Advisor：Yu-Chee Tseng

國 立 交 通 大 學
資 訊 科 學 與 工 程 研 究 所
博 士 論 文

A Dissertation
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science

October 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年十月

藉由無線感測網路來設計與實做一套智慧型生活空間

學生：葉倫武　　　　　　　　　　　　　　　　指導教授：曾煜棋教授

國立交通大學資訊科學與工程研究所

摘　　　要

近年來，無線感測網路已廣泛被運用在許多方面，而智慧型生活空間是其中一項很重要的應用。本篇論文共包含四項關於智慧型生活空間的研究主題。第一項研究主題是一套智慧型的省電系統，其建構出一套智慧型生活空間的架構，而在這套架構裡，我們提出了兩項靜態的服務與一項動態的服務，此兩項靜態的服務包含了燈源控制系統與空調控制系統，此兩項系統能夠滿足使用者需求並達到節能的目的，而動態服務方面則是提供一套以地理資訊為基礎之服務平台。

在第一個研究主題方面，我們利用無線感測網路提出了一套智慧型個人化節能系統，此系統包含了無線感測網路與家電控制來提供個人化的節能服務。藉由遍佈在環境中的無線感測器來監控室內環境，用以決定何時可關閉該電器，來達到省電的效益。此套系統可以根據使用者的個別需求來自動的調整電器，藉以滿足使用者個人化的需求。

在第二個研究主題方面，我們提出了一套自主式燈光調控系統，此系統藉由使用者所攜帶的光度感測器，來回報目前光度值，用以做為調控燈具之依據。本系統主要是以考量使用者需求與節能為目標，在燈源方面，同時考慮到了全區與局部燈源，而在使用者需求方面，考量了兩種模型，分別以最小化用電量為目標或是以最大化使用者滿意度為目標兩種。我們提出了兩種控制全區燈源的演算法，與一套表面追蹤的演算法用於局部燈源。相較於其他燈源控制系統，當環境改變時，我們的系統可以動態的自動調整所需的光源，並且無需追蹤使用者位置。而在系統驗證方面，除了模擬之外，我們還實作出整套系統，驗證本系統之可行性。

而在第三個研究主題裡，我們提出了一套智慧型的空調控制系統，此系統藉由環境中的溫度感測器，來回報環境中的溫度，用以做為調控的依據。本系統主要是以考量使用者需求與節能為目標，對此我們考量兩種模型，分別以最小化用電量為目標或是以最大化使用者滿意度為目標兩種。對此兩種目標，我們建構出一套溫度模型，與兩套空調控制演算法，用以快速的調整空調系統，來滿足使用者溫度需求。

在最後一個研究主題裡，我們對於居家或辦公環境裡，建構了一套低成本的移動平台。此移動平台可自主式的移動並決定其所在位置，用以提供以地理位置為基礎之相關服務。對於此移動與定位方式，我們提出了一套兩層式架構，讓此移動平台可以達到公分級與公尺級的位置精確度，故此移動平台可以移動到環境中的任意點，來提供使用者所需的服務。我們利用 iRobot 來實做出整套系統，並驗證其可行性。

關鍵字：情境感知、節能、智慧型建築、室內定位、LED、燈源控制、普及運算、RFID、機器人、感測網路、智慧型環境、無線網路、無線感測網路。

Design and Implementation of a Smart Living Space by Wireless Sensor Networks

Student: Lun-Wu Yeh Advisor: Prof. Yu-Chee Tseng

Department of Computer Science

National Chiao Tung University

ABSTRACT

Recently, wireless sensor and actuator networks (WSANs) have been widely discussed in many applications. The smart living space is one of the most important application of WSANs. In this dissertation, we propose four works for a smart living space. The first work is an intelligent energy-conservation system which constructs a framework of the smart living space. Under this framework, we propose two static services and one mobile service in this environment. Two static services can control lighting devices and air conditioners automatically to achieve energy saving and satisfy users' requirements. One mobile service provides a location-based services platform in our environment.

In the first work, we propose an *iPower (intelligent and personalized energy-conservation system by wireless sensor networks)* system which combines WSNs and appliance control devices to provide personalized energy conservation services. A WSN is deployed in each room to monitor the usage of electric appliances and to help determine if there are electric appliances that can be turned off for energy conservation. The iPower system is quite intelligent and can adapt to personal need by automatically adjusting electric appliances to satisfy users' requirements.

In the second work, we propose an autonomous light control system based on the feedback from light sensors carried by users. Our design focuses on meeting users' preferences and energy efficiency. Both whole and local lighting devices are considered. Users' preferences may depend on their activities and profiles and two requirement models are considered: binary satisfaction and continuous satisfaction models. For controlling whole lighting devices, two decision algorithms are proposed. For controlling local lighting devices, a surface-tracking scheme is proposed. Our solutions are autonomous because, as opposed to existing solutions, they can dynamically adapt to environment changes and do not need to track users' current

locations. Simulations and prototyping results are presented to verify the effectiveness of our designs.

Similar to the previous work, we propose an intelligent air conditioners control system based on the feedback from temperature sensors in the environment. Our design focuses on meeting users' requirements and energy efficiency. We define two models as users' requirements: binary satisfaction and continuous satisfaction models. For these two models, we propose the temperature model and two control mechanisms to adjust the air conditioners quickly and satisfy all users' temperature requirements.

In the last work, we consider building an indoor low-cost mobile robot that can be used in home applications. Due to the complicated nature of home environments, it is essential for such a robot to be *self-guided* in the sense that it is able to determine its current location as well as navigate to locations where it is commanded to. We propose a two-tier architecture to achieve this goal at centimeter-to-meter-level accuracy. The robot can even roam into an area which is new to it. We demonstrate a prototyping system based on an extended iRobot and the results have important implications on intelligent homes.

Keywords: context awareness, energy conservation, intelligent building, indoor positioning, LED, light control, localization, pervasive computing, RFID, robot, sensor network, smart environment, wireless communication, wireless sensor and actuator network.

Acknowledgement

Special thanks goes to my advisor Prof. Yu-Chee Tseng for his guidance in my dissertation work. I would also like to thank my dissertation committee members: Prof. Chih-Wei Yi, Prof. Chien Chen, Prof. Sheng-Tzong Cheng, Prof. Jia-Shung Wang, Dr. Wen-Tsung Chang, Dr. Shiaw-Shian Yu, Dr. Yueh-Feng Lee. They asked me some good questions and gave me useful comments so that I can improve my work in the future.

Let me also say thank to those HSCC (High-Speed Communication & Computing Laboratory) members who co-work with me and all guys I meet in NCTU. Because of you, I can have a great time during these years. Finally, I will dedicate this dissertation to my families for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid progress of wireless communications and embedded MEMS technologies has made *wireless sensor and actuator networks* (WSANs) possible. A WSAN [19][20][61] is a distributed system consisting of sensor and actuator nodes interconnected by wireless links. Using sensed data from sensor nodes, actuators can perform actions accordingly. The possible applications of WSANs include smart living spaces [45][62], localization [31][43], environmental monitoring [41][64], etc.

The smart living space is one of the most important application of WSANs. Through the sensory data which are reported by sensor nodes in the environment, some actuators, such as lamp, air conditioner, dehumidifier, etc, can be automatically controlled. There are many products about controlling appliances [24][57] automatically. However, these products only can be turned on, turned off, or even adjusted the level of the appliances. They do not concern the users' preferences and activities. Hence, we want to propose several systems which not only can control appliances automatically but also consider users' preferences.

In this dissertation, we propose four works about the smart living space by wireless sensor

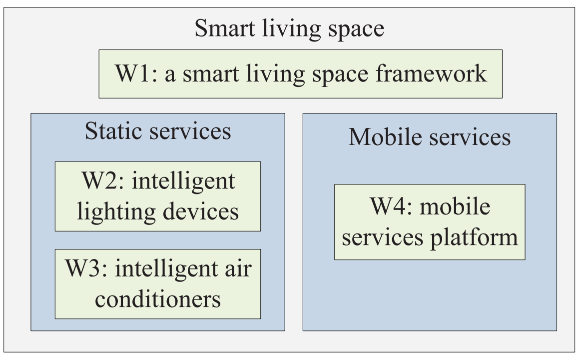


Figure 1.1: The organization of the dissertation.

networks. As shown in Fig. 1.1, the dissertation is composed of four parts. The first one is an energy conservation system which constructs a framework of the smart living space. Under this framework, we propose two static services and one mobile service in the environment. Two static services can control lighting devices and air conditioners automatically to achieve energy saving and satisfy users' requirements. One mobile service provides a location-based services platform in our environment.

In the first work, we propose an *iPower (intelligent and personalized energy-conservation system by wireless sensor networks)* system which combines WSNs and appliance control devices to provide personalized energy conservation services. The system can exploit the context information by WSN. The WSN is deployed in each room to monitor the usage of electric appliances and to help determine if there are electric appliances that can be turned off for energy conservation. The iPower system is quite intelligent and can adapt to personal need by automatically adjusting electric appliances to satisfy users' requirements. The iPower system construct a framework of the smart living space. Under this framework, we construct following three systems in the smart living space.

The second work is an autonomous light control system. The light control system based on the feedback from light sensors carried by users. Our design focuses on meeting users' preferences and energy efficiency. Both *whole* and *local lighting devices* are considered. Users' preferences may depend on their activities and profiles and two lighting requirement models are considered: binary satisfaction and continuous satisfaction models. For controlling whole lighting devices, two decision algorithms are proposed. For controlling local lighting devices, a surface-tracking scheme is proposed. Our solutions are autonomous because, as opposed to existing solutions, they can dynamically adapt to environment changes and do not need to track users' current locations.

Similar to the light control system, we design an energy saving air conditioners control system based on the feedback from temperature sensors in the environment. Our design focuses on meeting users' requirements and energy efficiency. We define two models as users' temperature requirements: binary satisfaction and continuous satisfaction models. For these two models, we propose the temperature model and two control mechanisms to adjust the air conditioners quickly and satisfy all users' temperature requirements.

In the last work, we consider building a mobile services platform by an indoor low-cost mobile robot that can be used in home or office applications. Due to the frequently changed and complicated nature of a home/office environment, it is essential for such a robot to be *self-guided* in the sense that it is able to determine its current location as well as navigate to locations where it is commanded to. By integrating with wireless communication, indoor localization, and RFID technologies, it is possible to build a self-guided robot at low cost. We propose a two-tier 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. On the first tier, it determines its rough location via WiFi localization at meter-level accuracy. On the second tier, it then tries to track any RFID tag registered in the indoor map; on events of finding any tag, it actually localizes itself at centimeter-level accuracy.

To examine the performance of our designs, we not only verify by simulation programs but also build the prototyping results in real world. We implement the light control system including whole and local lighting devices and the mobile services platform in a room. For whole lighting devices, we build the whole lighting devices in a grid manner to test the performance of two decision algorithms. For local lighting devices, we construct a robot arm holding four sets of LEDs and an electric bookmark to experiment our proposed algorithm. For mobile services platform, we adopt iRobot Create [25] as the mobile platform and attach a RFID reader, an electronic compass, and a notebook with WiFi interface to verify the self-guided mechanism of the robot.

The major contributions of this dissertation are summarized as follows.

- Framework: We propose a smart living space framework which contains two static and one mobile services.

- User requirement: For light and air conditioner control systems, we consider each user may have different lighting and temperature requirement.

- Without location information: Our energy saving light control mechanism can adjust lighting devices to satisfy users' requirements without extra indoor localization scheme to track users' locations. We only use the "light intensity" to get the information that we need.

- Localization by light intensity: For local lighting devices, our mechanism can locate object in centimeter-level precision by light intensity.

- Quick control: Air conditioner control system can adjust the control level more quickly than PID method.

- Self-guided robot: We propose a two-level self-guided mechanism for indoor mobile robot navigation.

This dissertation is organized as follows. Chapter 2 presents a smart living space framework. In Chapter 3, intelligent lighting and air conditioners control systems are proposed. Chapter 4 presents the indoor mobile services platform. Finally, we conclude our results and propose some future directions in Chapter 5.

Chapter 2

A Smart Living Space Framework

The discovery of electricity is one of the most important milestones in human history. Electricity is so essential in our daily life that many people cannot live without it. However, today, energy has been overly used and energy shortage has become a global concern. According to the report in [18], more than one third of electricity is spent on *HVAC systems*, which include heating, ventilating, air conditioning, lighting, and other related equipments. According to experiences, a large portion of energy consumed by HVAC systems is due to improper use of electric appliances. Therefore, how to exploit the context information of an environment to automatically control the usage of electric appliances has a great potential to reduce the waste of energy.

2.1 System Overview

In this chapter, we therefore propose an *intelligent and personalized energy-conservation system by wireless sensor networks (iPower)* to reduce energy consumption of HVAC systems by exploiting the context-aware capability of sensors. In the iPower system, WSNs are deployed in rooms of a building to collect information of the environment. Such information is reported to a control server to determine whether to turn off those unnecessary electric appliances in the building. Such a system needs to be designed with user friendliness in mind to minimize the involvement of users in making decisions. As an example, when sensor nodes detect a low temperature or a high brightness in a likely unoccupied room, they can report to the server that the electric appliances in that room (e.g., air conditioners or lights) could be turned off. The server then sends an alarm signal to notify people in the room that the electric appliances could be turned off shortly. If there are still users in that room, they can signal the system that these appliances should not be turned off by triggering some events (such as making some voices, changing the light reading of any sensor, or moving any furniture attached with sensors). If there is no such intentional events made by human being detected in a predefined amount of time, the server will turn off the electric appliances through some power-line control devices. In this way, the iPower system can work even if users are not wearing any particular badge.

In the iPower system, we also provide *personalized services* in which electric appliances can be automatically adjusted to satisfy users' preferences. In particular, each user can create a profile to describe his/her favorite temperature and brightness. Such users are considered priority users and need to carry portable sensors so that our system can retrieve their profiles. When there are priority users in a room, the server will adjust the air conditioners and lights in that room according to the profiles of these users.

WSN has been widely used to provide context information in smart environment. How to automatically control electric appliances according to users' locations and their requirements has been intensively discussed for smart homes/offices. The work in [51] considers a ubiquitous computing architecture in which electric appliances are controlled by a SIP (session initiation protocol) [56] server, under which architecture users can make calls to communicate with the SIP server to control their electric appliances. In the *MavHome* system [14], the mobility pattern of a user in a house is exploited and is forwarded to the system to provide advanced services (e.g., controlling the corresponding electric appliances) in the predicted locations of the user. In *Semantic Space* [62], the authors propose some semantics to describe the environment, which can be used to query the status of the environment where users are located. The work in [21] proposes a context-aware smart house in which electric appliances can be automatically adjusted according to the environmental information collected from sensors. The iPower system is motivated by observing that the issue of energy conservation, which is very critical to our environment, has not been well addressed.

2.2 Design of the iPower System

2.2.1 System Architecture

The architecture of our iPower system is illustrated in Fig. 2.1, which consists of many sensor nodes, several sink nodes, an intelligent control server, some actuators, and portable sensors. Below, we describe the functions of each component separately

- **Sensor nodes:** In each room, we deploy sensor nodes to monitor the environment. These

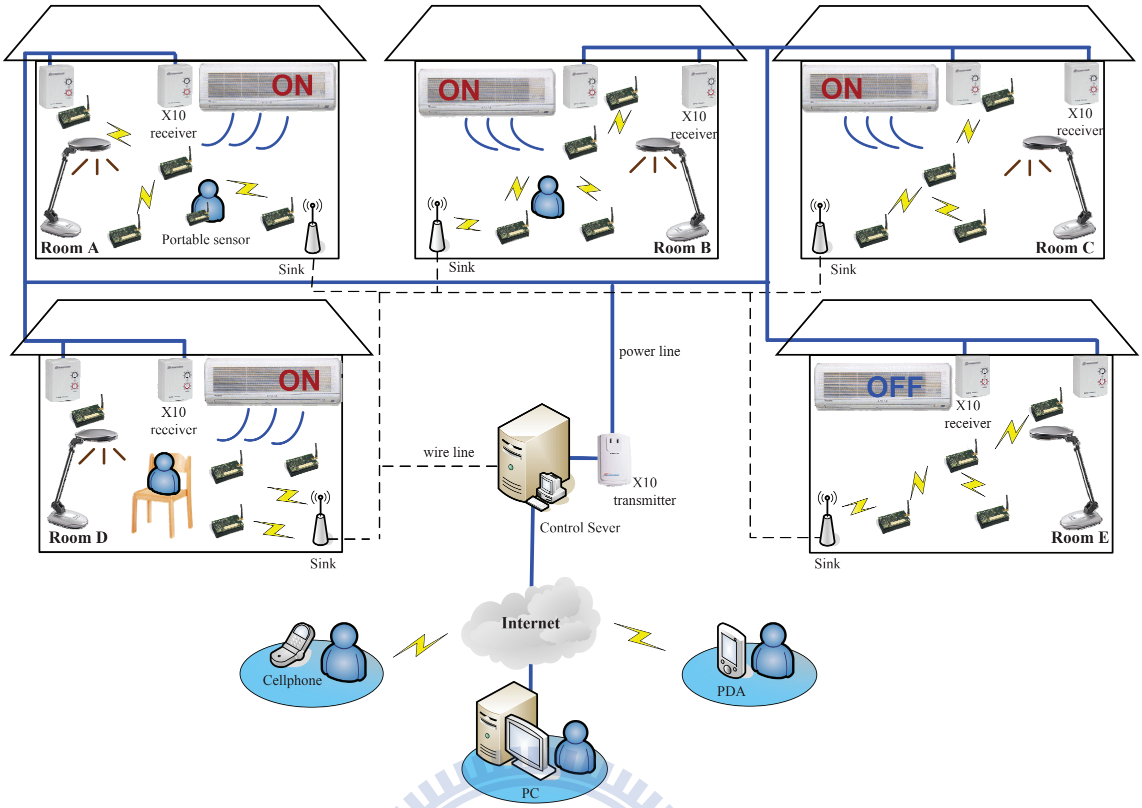


Figure 2.1: System Architecture of the iPower.

nodes will form multi-hop WSNs to collect information in the rooms. In our current prototype, three types of sensing data can be collected, including light, sound, and temperature. An event is defined when the sensory input is higher or lower than a predefined threshold. To conserve the energy of sensor nodes, reporting of events is reactive, in the sense that a node will report its sensing data only when some predefined events occur. Different events can be combined to describe a room's condition. For example, a low temperature (or a high brightness) together with some sound events in a room may indicate that the corresponding electrical appliances are turned on to serve users in that room; some sound events and change of the light degree read by sensors may indicate that users in that room are moving around; and a low temperature (or a high brightness) with no sound event for a certain amount of time may indicate that the air conditioners (or lights) in that room are unnecessarily turned on because no one is in the room. We can include more types of sensors to provide more intelligence. For example, as shown in Fig. 2.2, a smart desk may include some pressure sensors underneath the cushion of a chair and some light sensors nearby the lamp on the desk. When someone is sitting on the chair, such an event can be detected by the pressure sensors, and the system can adjust the lamp according to the light degree nearby the lamp. When the user leaves the chair, the pressure sensors can detect the disappearance of the user and make energy conservation decision by notifying the server to turn off the lamp.

- **Sink nodes:** The set of sensor nodes in each room will form a WSN. For each WSN, there is a sink node. A sink node has a wireless interface to communicate with sensor nodes and a wire-line interface to communicate with the intelligent control server. It has four major functionalities: issuing commands to sensor nodes, gathering data from sensor nodes, reporting the room's condition to the intelligent control server, and maintaining the WSN. Specifically, the sink will notify sensor nodes in the WSN to begin collecting environmental information when it receives a *start* command from the server. After gathering sensing reports from the WSN, the sink node will determine the room's condition and report to the server. In order to maintain the WSN, the sink node will periodically broadcast a *heart-beat* message to the network. A sensor node receiving such a message will reply an *alive* message to the sink node. If the sink node does not receive any *alive* message from a sensor node for a predefined amount of time, it will notify the server that the node may be broken.

- **Control server:** The control server is used to collect the system's status (e.g., rooms' conditions and sensors' states) and to perform power-saving decisions. It maintains a database of user profiles and periodically checks the states of electric appliances in each room. It will decide whether to turn off an electric appliance in a room according to the sensory data collected from that room. The server can also adjust the electric appliances in a room according to the profiles of users in that room. Such decisions or adjustments are achieved by sending commands through the power-line control devices to turn off or adjust electric currents of the corresponding electric appliances. The server also provides user interfaces to allow users to maintain the iPower system. In particular, users can modify their profiles and obtain the system's status through remote devices.

- **Actuators:** In iPower system, we control the actuators by the power-line control devices. The power-line control devices allow the system to turn on/off or adjust the electric currents of appliances. In our current prototype, we adopt the X10 devices produced by SmartHome [57]. Such devices contain one X10 transmitter and several X10 receivers.

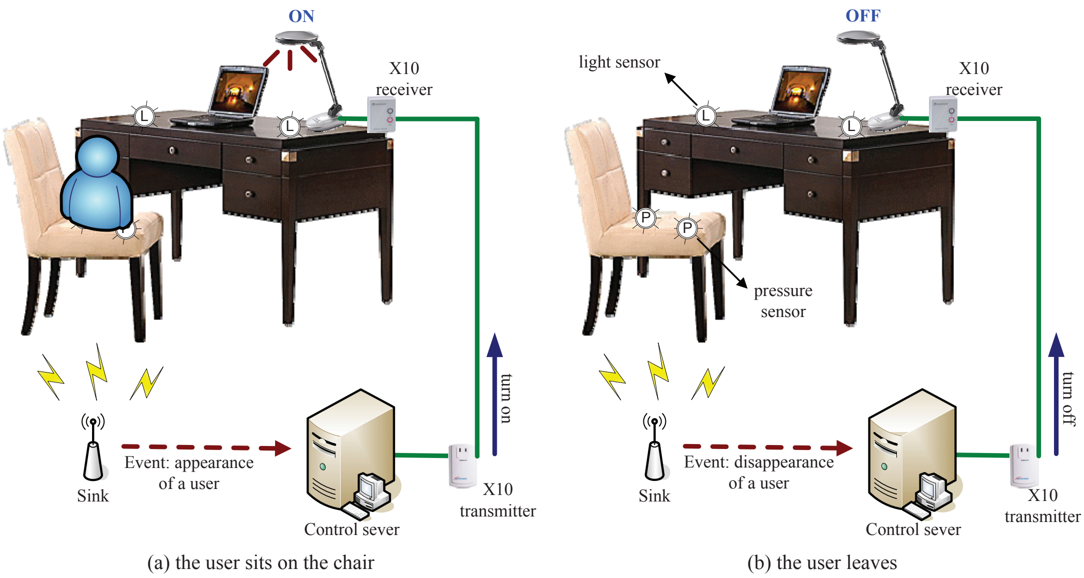


Figure 2.2: The "smart desk" scenario.

The X10 transmitter can talk to X10 receivers via power lines. In the iPower system, the X10 transmitter is connected to the control server to transmit the server's commands.

- **Portable sensors:** The portable sensors are the portable devices that can be carried by users so that the system can determine users' IDs and retrieve their profiles. It can be any identification device. In this work, we simply use the processor board of our sensor platform (without sensors) for user identification. When a user enters a room, his/her user portable sensor will join the WSN in that room and provide its ID to the server via the sink node.

2.2.2 Energy Conservation Scenarios

Next, we give five scenarios to demonstrate how the iPower system works in an intelligent building. Let us consider the five rooms in Fig. 2.1.

- **Room *A*: electric appliances are turned on and somebody is in the room (with a portable sensor).** In this case, since the system can detect that the room is occupied, energy conservation commands will not be issued. So the electric appliances in room A will remain on.

- **Room *B*: electric appliances are turned on and somebody is in the room (without a portable sensor).** In this case, energy conservation commands will be given depending on whether some events (such as sound events) indicating that the room is occupied can be detected or not. If there are such events, the electric appliances will remain on. Without such events, some signals (such as beeps or blinking lights) will be triggered to warn users in that room. In response, users can do some actions to signal the system that the room is occupied (such as making some noise by clapping, covering any sensor with a light sensor to change its light reading, or switching on or off any electrical appliance that is under control of the iPower system). As long as any of such events can be detected, the server can realize that the room is still in use and thus will not turn off the electric appliances. Note that to reduce bothering users too much, the interval to warn users next time will be increased in an exponential manner after each intentional event being generated by users in that room. Further, after several warning signals without success, the system will stop trying (to make energy conservation decisions) for a long period of time.

- **Room *C*: electric appliances are turned on but nobody is in the room.** In this case, since sensor nodes have detected a low temperature, a high brightness, and no sound event for some while, the sink node will report to the control server that this room is *abnormal*, implying that electricity may be wasted in room C. The server will then send an *alarm* message to room C, which triggers the beepers attached to sensor nodes. These beeps are used to announce that the system will turn off air conditioners and lights in room C in a few minutes. Alternatively, we can blink lights on and off to signal users that appliances in that room will be turned off soon. This is to avoid our system to make wrong decisions. Since there is no one in the room, the server will turn off these appliances after timeout to conserve energy.

- **Room *D*: electric appliances are turned on in the room with smart furniture.** If there is smart furniture in the room, they can help detect the existence of people in that room. For example, if there is a person sitting on a smart chair, the system will keep on reporting that someone is on the chair, so no energy conservation commands will be issued. If the smart furniture is not in use, then the scenario in room B may be applied.

- **Room *E*: electric appliances are turned off.** In this case, the sink node will report to the server that the room is normal so the server will not take any action.

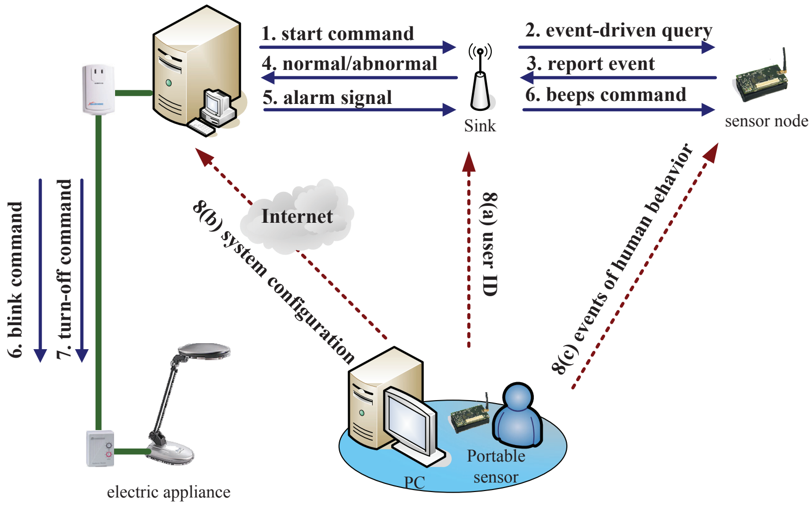


Figure 2.3: Message flows in the iPower system.

2.2.3 System Operations and Message Flows

Fig. 2.3 illustrates the message flow and the interaction of system components in the iPower system. The details are discussed below.

1. The control server starts checking the usage of electric appliances in a room by sending a *start* message to the sink node in that room. Checking can be done periodically or at predefined time, according to the system configuration file.

2. On receiving the *start* message from the server, the sink node will notify its sensor nodes by issuing some event-driven queries to collect information from the environment. The sink node then sets a timer to wait for sensing reports from sensor nodes.

3. When a sensor node detects any event (such as a low temperature or a high brightness), it will report its sensing data to the sink node.

4. If the sink node receives any sensing report and any human behavior report from step 3 before its timer expires, it can determine the room's status according to the following rules:

    (a) If any piece of smart furniture reports that someone is using it (e.g., the case in Fig. 2.2(a)), the sink node will report a *normal* status to the server. However, if it is reported that users leave the smart furniture, the sink node will reset its timer and go back to step 2 to repeat the aforementioned procedure.

(b) If sensors report any human behavior (such as sound events or change of light readings), the sink node will report a *normal* status to the server. However, it will also notify the existence of people to the server so that the system will check this room's status later on.

(c) Otherwise, the sink node will report an *abnormal* status to the server to indicate that the electric appliances in the room may be turned on unnecessarily.

5. When the server receives an *abnormal* report from the sink node, it will warn the people (if any) in the corresponding room by sending an *alarm* message to the sink node.

6. Once receiving the *alarm* message, the sink node will instruct one of its sensor nodes to turn on its buzzer to generate a beeping sound. Alternatively, the server can send a *blink* command to the X10 receiver to blink any light on and off for a short period of time. These actions are used to notify people in the room that the server will turn off the electric appliances after a short period of time (e.g., ten minutes).

7. If the server does not receive any human behavior event from the room after a predefined period of time, it will know that there is no one in that room and thus turns off the electric appliances by sending a *turn-off* command to the X10 receivers in that room.

8. If there is any user in the room hearing the beeping sound or seeing blinking light, he/she can notify the server that the room is still in use by any of the following three methods:

   (a) If the user has carried a portable sensor, the device will directly inform the server (via the sink node) his/her ID. In this case, the user does not need to take any action.

   (b) If the user can access the Internet, he/she can login the web page of the iPower system to set up the next checking time of this room so that the server will not disturb the user before he/she leaves the room.

   (c) (a) Otherwise, the user can make some intentional events by changing the room's environment, such as making some noise by clapping or turning off and then turning on any light. In this way, sensor nodes will detect an unusual sound or change of light degree and thus report these events to the sink node.

According to these reports, the sink node can notify the existence of users to the server and thus the system will back off and check the room's status later on. The next checking time can

```xml
<? xml version="1.0" encoding="UTF-8" ?>
- <User>
    <Id>007</Id>
    <Name>HSCC</Name>
  - <Attribute>
      <Name>temperature</Name>
      <Value type="Float"/>
      <Range max="28" min="25">true</Range>
    </Attribute>
  - <Attribute>
      <Name>light</Name>
      <Value type="Float">70</Value>
      <Range max="0" min="0">false</Range>
    </Attribute>
  </User>
```

Figure 2.4: An example of user profile.

be set manually by users, by any default value (such as one hour), or in any typical exponential backoff manner.

2.2.4 Personalized Services and User Profiles

The iPower system also provides personalized services in which electric appliances can be automatically adjusted to satisfy users' preference. In particular, each user can specify his/her favorite temperature and brightness. When a user enters a room, the iPower system can adjust the air conditioners and lights to meet the user's preference. To achieve this goal, the user has to create a profile in the server's database and carry a portable sensor when entering our system. The user's location is determined by the sink node which collects the user's ID. In our current implementation, we follow the format of *XML (extensible markup language)* [66] to describe user's profiles. The current definition is illustrated in Fig. 2.4. Specifically, the profile includes user's ID, name, and several attributes with the user's favorite temperature and brightness. For example, Fig. 2.4 indicates that user's preference temperature is from $25\,^{\circ}\mathrm{C} \sim 28\,^{\circ}\mathrm{C}$ and light is 70 lux.

2.2.5 Events and Actions

One of the main components of iPower is its automatic rules. A *rule* can be composed of time, events and actions. A rule can be event-driven or time-driven. Actions can be triggered by *simple events* or *compound events*, where the latter are combinations of multiple simple events. For example, when someone is sitting on a smart chair near a smart desk with a low light degree, to automatically turn on the lamp on the desk, we need to combine events from pressure sensors and light sensors. Note that compound events can be combined through logical operations, such

```
<iPowerRule>   = [ User <UserID> ] On <Time> Condition <Event> Do <Action>
<UserID>       = string
<Time>         = min/hr/date/mon/yr | anytime
<Event>        = <Sensor> AND <Event> | <Sensor> OR <Event> |
                 <DeviceStatus> AND <Event> |  <DeviceStatus> OR <Event> |
                 <Sensor> | <DeviceStatus>
<Sensor>       = <SensorID> <SenseData>
<SensorID>     = string
<SenseData>    = <SenseType> <Range>
<SenseType>    = temperature | sound | pressure | humidity | light | rssi
<Range>        = <Max> To <Min>
<Max>          = integer | float
<Min>          = integer | float
<DeviceStatus> = <DeviceID> <DeviceInfo>
<Action>       = <Device> AND <Action> | <Device>
<Device>       = <DeviceID> <DeviceInfo>
<DeviceID>     = string
<DeviceInfo>   = <DeviceAction> AND <DeviceValue> | <DeviceAction>
<DeviceAction> = on | off
<DeviceValue>  = integer | float
```

Figure 2.5: The iPower’s EBNF-like recursive grammars.

```
Room A:
  User userID_1
  On anytime
  Condition ( sensorID_1 temperature 28 To 50 )
  Do device_aircon on
Room C:
  On anytime
  Condition ( sensorID_2 rssi 0 To 40 ) AND ( sensorID_2 rssi 80 To 100 ) AND
            ( device_lamp on )
  Do device_lamp off
Room D:
  On anytime
  Condition ( sensorID_3 light 0 To 20 ) AND ( sensorID_4 pressure 10 To 100 )
  Do device_lamp on
```

Figure 2.6: Examples of the iPower’s rule.

as “AND” and “OR”.

In Fig. 2.5, we list the definition of iPower’s rule, which are written in the format of *EBNF (Extended Backus-Naur Form)* [52] recursive grammar. Each iPower’s rule defines for a certain *User*, when some *Time* and some *Conditions* are matched, the corresponding actions to be taken. Terms quoted by $[\cdots]$ are optional. For example, when $<$*UserID*$>$ in a rule is not specified, it means anyone can match this rule. Fig. 2.6 shows the rules for rooms A, C and D in Fig. 2.1. Note that here we use *RSSI (received signal strength index)* between 40 and 80 to indicate that a user’s badge in within the range of a WSN.

2.2.6 Protocol Stack

To implement the iPower system, we have designed a protocol stack in Fig. 2.7, which consists of the following modules:

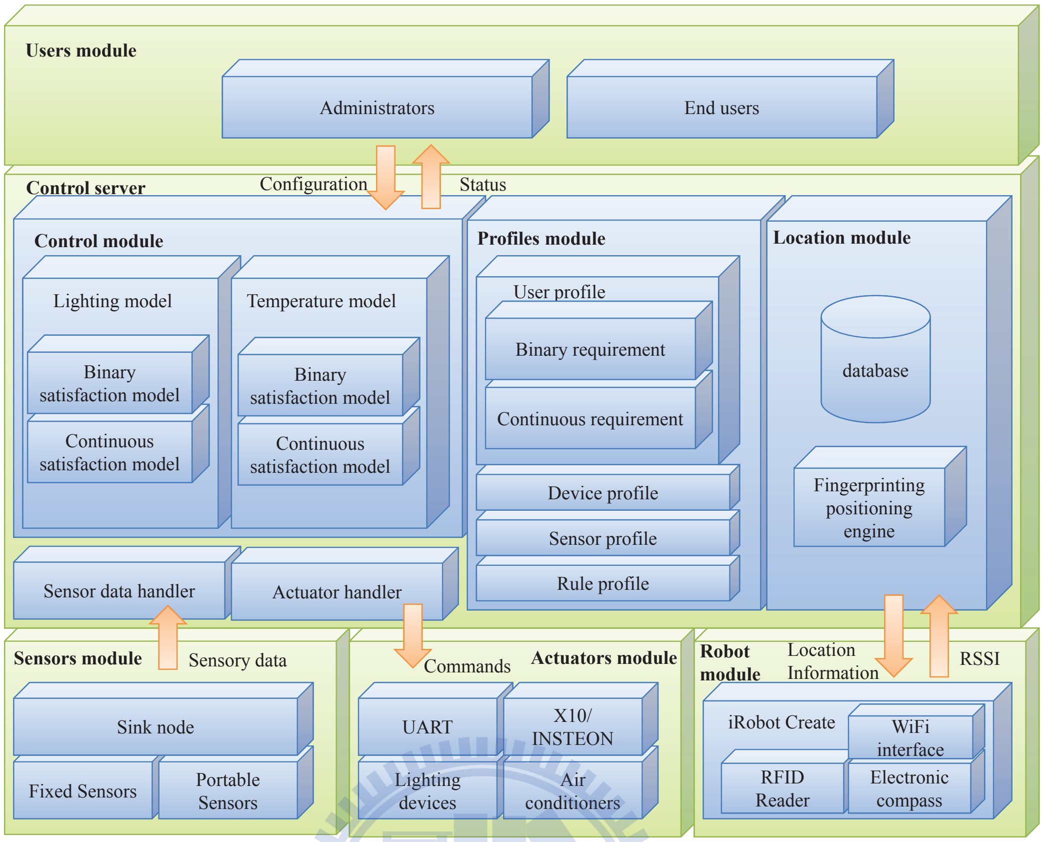


Figure 2.7: Protocol stack of the iPower system.

- **Users module**: The users module defines how a user can access the system through the user interface. Here we consider two kinds of users: *administrators* and *end users*. An administrator can add or remove equipments (e.g., electric appliances, sensors, and power-line control devices) in the system, change their attributes and profiles, and manage end users. An end user can only create and modify his/her user profile.

- **Profiles module:** The profiles module maintains all profiles for users, sensors, devices, and rules. The *sensor profiles* describe the locations and sensing types of sensors. The *device profiles* describe the electric appliances controlled by the power-line control or UART devices. The *user profile* contains two kinds of users' requirements: binary and continuous requirements. These two requirements are used for two models in lighting and temperature models. Finally, the *rule profiles* define how the components in the iPower system interact with each other.

- **Sensors module**: The sensors module controls the actions of sensor nodes. These actions include executing commands from the the sink node (such as to detect events and to generate beeping sounds) and reporting sensing data to the sink node.

- **Actuators module**: This module provides an abstraction of electric appliances to upper layers. In our implementation, we choose X10, INSTEON [24] and UART as our device control protocols. Through these protocols, we can turn on, turn off, and adjust the electric currents of appliances.

- **Robot module**: The robot module provides a mobile service platform. It is composed of iRobot Create [25], RFID reader, electronic compass, and WiFi interface device. The details of mobile service platform are presented in Chapter 4.

- **Location module**: The location module contains the fingerprinting positioning engine and a location information database. The location module provides the location and area information for robot module.

- **Control module**: The control module contains lighting and temperature models. Through these two models, the control server can decide the control levels of lighting devices and air conditioners. The details of light and temperature control are presented in Section 3.3 and Section 3.4.

2.3 Implementation Details

2.3.1 Hardware Specification

We use the Jennic (JN5139) [26] as sensor nodes. The Jennic is a 2.4 GHz, IEEE 802.15.4-compliant module that enables low-power operations and offers a data rate of 250 kbps with a DSSS radio. Each sensor node has a sensing board that can collect sensing data from their surrounding environment, including light, sound, and temperature. More sensors can be added on the board to increase the sensing capabilities. Each sensor node also has a buzzer to generate a beeping sound when they are commanded by the sink node.

For the power-line control devices, we adopt the X10 products by SmartHome. The X10 devices consist of X10 transmitters and X10 receivers. They can communicate with each other by the X10 communication protocol, which encodes messages on the electric signal with a frequency of 60 Hz. With the X10 communication protocol, an X10 transmitter can send commands to an X10 receiver through a power line. To control electric appliances, we connect one X10 transmitter to the server via an RS-232 interface and connect all electric appliances

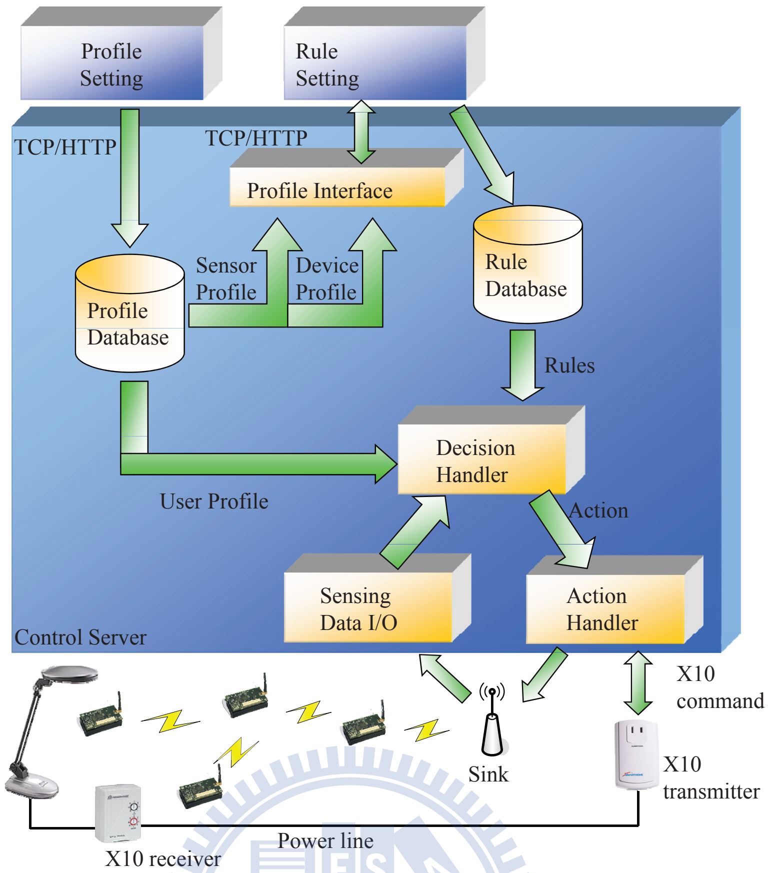


Figure 2.8: Design of the control server.

with X10 receivers. Each X10 receiver has a unique address and at most 256 addresses can be selected.

2.3.2 Design of the Control Server

The control server is the core of our iPower system. Fig. 2.8 illustrates the design of the server. The implantation details are discussed below.

1. An administrator can add a sensor profile or a device profile through the *profile setting* component. Related information such as sensing types and device attributes can be created in the profile database.

2. An administrator can interact with the *profile interface* to create rules through the *rule setting* component.

3. A sink node can report environment information through the *sensing data I/O* interface.

4. The *decision handler* combines the user profiles, rules, and sensing data to generate proper actions.

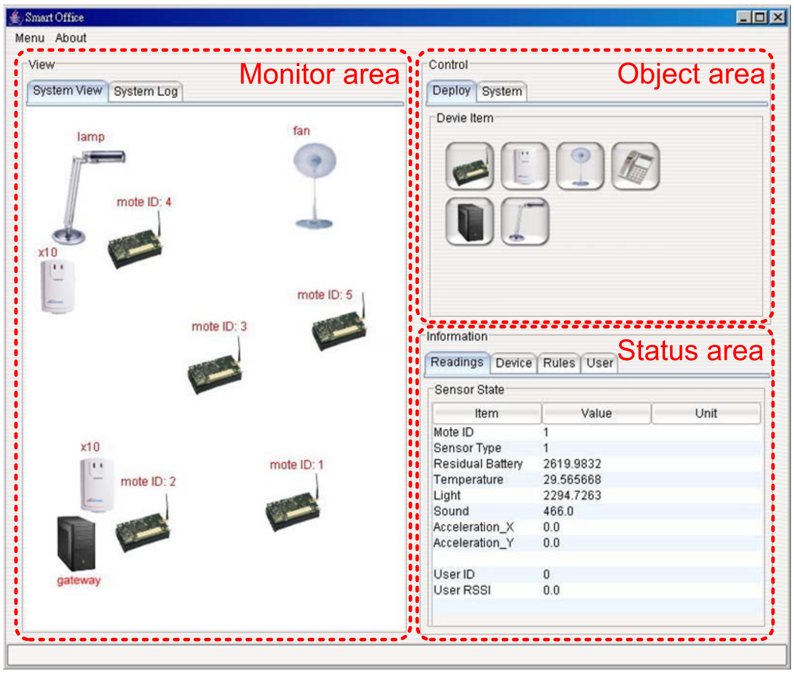


Figure 2.9: The user interface at the control server.

The actions are sent to the *action handler*, which can generate commands to X10 devices or sensors.

2.3.3 User Interface

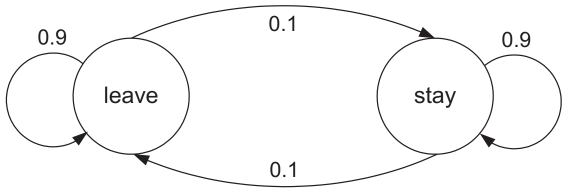
We provide a user interface to manage the system and allow users to create their profiles at the server, as shown in Fig. 2.9. The user interface has a *monitor area*, an *object area*, and a *status area*. The object area provides an interface to deploy all devices in a room, including sink node, sensor nodes, electric appliances, and X10 devices. This area also allows users to start or stop the system. In the monitor area, the administrator can visualize the deployment of sensors and devices. He/She can add new objects in the room by dragging objects from the object area to the monitor area. The monitor area also shows the network topology and electric appliance in the room. In the status area, the administrator can observe the attributes and the current status of each sensor.

2.4 System Evaluation

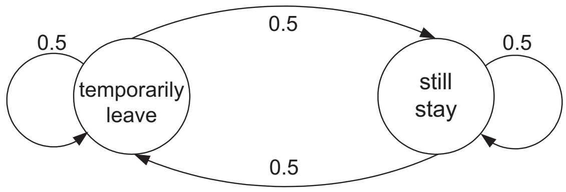
In this section, we present some simulation results to evaluate the system performance. We consider the energy consumption of an office with five people, one air conditioner, and five desk lamps, where each lamp is owned by one person. Table 2.1 lists the energy consumptions of

Table 2.1: Energy consumption of electric appliances.

electric appliance	energy consumption
air conditioner	800 watts/hour (at 28 °C)
desk lamp	80 watts/hour



(a) model a person's behavior during each hour



(b) model a person's detailed behavior during each twenty minutes

Figure 2.10: Two-state discrete Markov models.

different electric appliances. For the air conditioner, we assume that it will spend extra 100 watts when the temperature is decreased by 1 °C. A *two-state discrete Markov* model [29] is used to model a person's behavior during every hour, as shown in Fig. 2.10(a). A person can be either in one of the two states: *leave* or *stay*. When a person is in a leave state, the corresponding desktop lamp will be turned off. We use another Markov model to model the detailed behavior of a person when he/she in a stay state, as shown in Fig. 2.10(b). In particular, during every twenty minutes, the person may decide whether to "still stay" in the office or "temporarily leave" the office. When the person decides to temporarily leave the office, his/her own desktop lamp will remain on if the iPower system is not applied. Table 2.2 lists the favorite temperatures of the five people. When there are two or more people in the office, the temperature of the air conditioner will be adjusted to the average of favorite temperatures of those people in the office. Note that without iPower, we only adjust the temperature of the air conditioner when people enter the office.

Fig. 2.11 shows the energy consumption with five people during ten hours. We can observe that without iPower, the energy consumption of the office is always higher than 1500 watts, even when there is no person in the office (i.e., hours 3, 7, 9, and 10). This is because when all people temporarily leave the office, the air conditioner and some desk lamps are still turned on. On the other hand, the iPower system can detect such situation and thus properly turn off some

Table 2.2: Favorite temperatures of the five people.

person	favorite temperature
A	25 °C
B	27 °C
C	26 °C
D	22 °C
E	24 °C

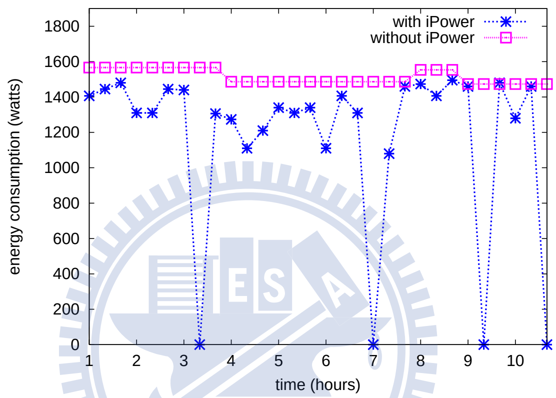


Figure 2.11: Energy consumption during 10 hours.

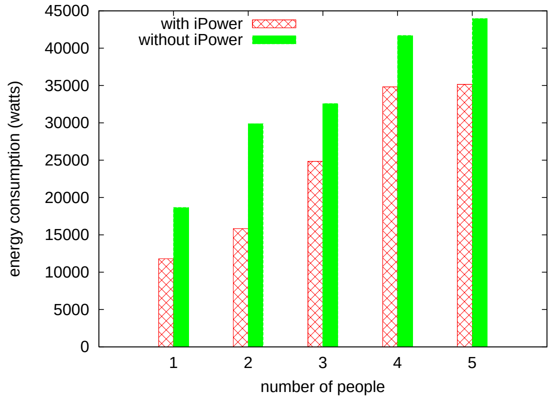


Figure 2.12: Total energy consumption with different numbers of people.

electric appliances to conserve energy. Fig. 2.12 compares the total energy consumption of the office with different number of people in the simulation. As can be seen, our iPower system can save approximate 16.5% $\sim$ 46.9% energy, which reflects its effectiveness.

Chapter 3

An Environment Control System for a Smart Living Space

3.1 System Overview

In this chapter, we propose an environment control system for a smart living space. The environment control system is composed of two sub-systems: light control and air conditioners control system. These two control systems consider both users' preferences and energy consumption. Fig. 3.1 shows our network scenario. Each user carries a portable sensor which can report current light and temperature readings to the sink node via wireless transmission. The fixed sensors, which are randomly deployed in the environment, also can sense and report current light and temperature readings to the sink node. Then the control server can give commands to lighting devices and air conditioners based on collected data. Here, the lighting devices and air conditioners are controllable by control server. The lighting devices are the LED lighting devices [32][34] and they are categorized into *whole* and *local lighting devices*. The former can provide background illuminations for multiple users in wide areas, and the latter is similar to desk lamps that provide concentrated illuminations. For example, in Fig. 3.1, device D_1 in the center can provide background illuminations for user u_1, u_2 and u_3, and device d_1 can only provide concentrated illumination for user u_3. In our system, users may have different illumination and temperature requirements according to their activities and profiles. Two types of lighting requirements, *background* and *concentrated* ones, are considered. For example, in Fig. 3.1, user u_2 is watching television, u_3 is reading a book, and u_4 is sleeping. Both u_2 and u_3 require the same background illuminations, but u_3 needs concentrated illumination. u_4 does not require neither background nor concentrated illuminations. A user is said to be *satisfied* if the provided current temperature, background illuminations, and concentrated illuminations

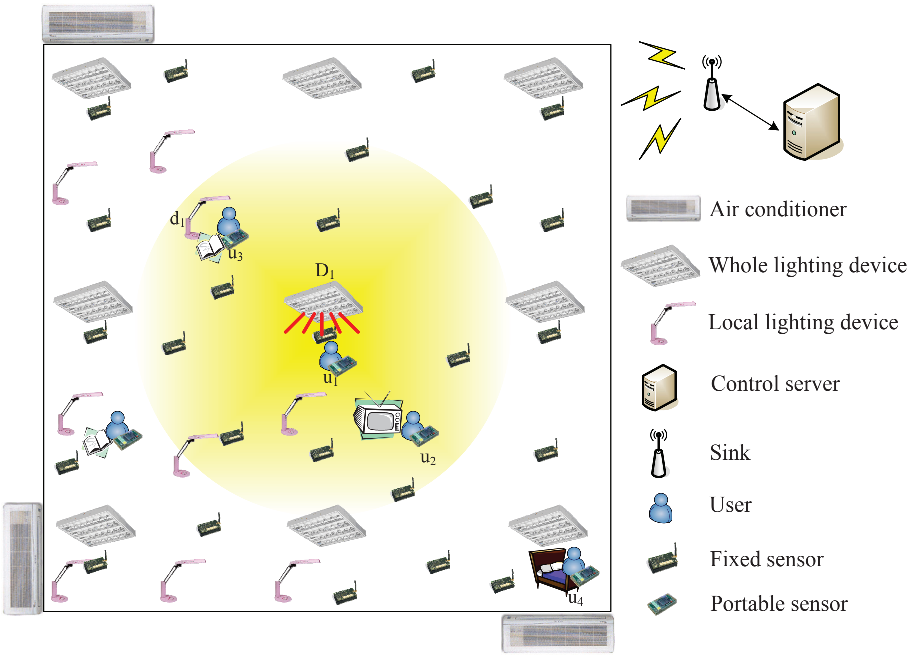


Figure 3.1: The network scenario of environment control system.

fall into the required ranges. To evaluate the satisfaction level of a user, we further consider a *binary satisfaction model* and a *continuous satisfaction model*. The former only returns a satisfaction value of 1 or 0, while the latter returns a value between 0 and 1. For light control, we develop two algorithms to adjust whole lighting devices for these models with the goals of meeting users' requirements while minimizing energy consumption. In case that it is impossible to satisfy all users simultaneously, we will gradually relax users' requirements until all users are satisfied. For concentrated illuminations, assuming that local lighting devices are moveable (which can be supported by robot arms), we develop a novel "surface-tracking" scheme to track local movements of users to provide required illuminations. For air conditioners control, we propose a temperature model and two air conditioners control algorithms to decide the proper temperature for air conditioners. The results not only satisfy all users' requirements but also achieve energy saving.

3.2 User Requirement Models for the Environment Control System

Here, we consider two kinds of user satisfaction models for lighting and temperature requirements:

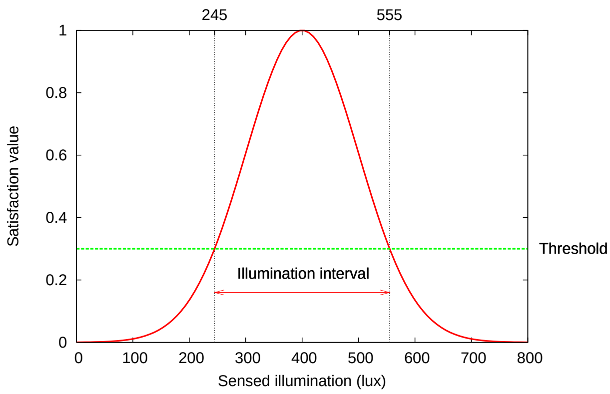


Figure 3.2: An example of continuous satisfaction.

1. *Binary satisfaction model*: For light control, each user u_i has an acceptable background illumination interval $[R_i^{bl}, R_i^{bu}]$ and an acceptable concentrated illumination interval $[R_i^{cl}, R_i^{cu}]$. The user is said to be *satisfied* if its concentrated and background illuminations fall within these intervals, respectively. Similarly, for air conditioners control, each user u_i has a temperature interval $[R_i^{tl}, R_i^{tu}]$.

2. *Continuous satisfaction model*: For light control, each user u_i has concentrated and background illumination requirements, but they are specified by utility-like functions[1]. The satisfaction value is given by a normal-distribution-like function with parameter μ_i, σ_i, and t_i. If x denotes the illumination, the satisfaction value $f_i^l(x)$ is defined as

$$f_i^l(x) = \begin{cases} exp(\frac{-(x-\mu_i)^2}{2(\sigma_i)^2}) & \text{, if} \quad x \in [\mu_i - \sigma_i\sqrt{-2ln(t_i)}, \mu_i + \sigma_i\sqrt{-2ln(t_i)}] \\ 0 & \text{, otherwise.} \end{cases} \quad (3.1)$$

   For the background illumination (or the concentrated illumination, respectively), we denote the three parameters as μ_i^b, σ_i^b and t_i^b (or μ_i^c, σ_i^c and t_i^c, respectively). Fig. 3.2 shows an example of continuous model with $\mu = 400$, $\sigma = 100$, and $t = 0.3$. Similarly, for air conditioners control, each user u_i has a temperature requirement function $f_i^t(x)$ with parameters μ_i^t, σ_i^t, and t_i^t. The definition of $f_i^t(x)$ is the same as $f_i^l(x)$.

[1]The work [55] adopts a curve function to represent users' lighting preferences. In this work, we adopt Gaussian functions to represent users' lighting and temperature preferences. However, it is not hard to extend to other utility functions.

3.3 An Intelligent Light Control System for a Smart Living Space

3.3.1 Motivation

Recently, WSANs have been applied to energy conservation applications such as light control [45][46][55][63]. The decision of lighting control can be made based on the daylight intensity sensed by light sensors [63]. In [46], the authors defined several user requirements and cost functions. Their goal was to adjust lights to minimize the total cost. However, the result was applied to entertainment and media production systems. In [55], a trade-off between energy consumption and user's satisfaction in light control was studied. The authors applied the utility functions which consider users' location and lighting preferences to adjust illuminations so as to maximize the total utilities. However, it did not consider the fact that people need different illuminations under different activities. The heuristics proposed in [46] and [55] need to measure all combinations of dimmer settings and the resulting illuminations at all locations, so if there are k interested locations, d dimmer levels, and m lighting devices, the measurement complexity is $O(kdm)$. In [45], for pervasive sensors deployment, the measuring time complexity was further reduced to $O(km)$. The goal was to satisfy users' demands while optimizing energy efficiency. These works all relied on knowing users' current locations, so extra localization mechanisms were needed.

The main contributions of this section are twofold. First, our model is designed for "point-like" light sources, such as LEDs, which are more energy-efficient than traditional light sources and are expected to be the mainstream of lighting technologies in the future. We show how to take advantage of its light propagation property to conduct light control. Second, compared to existing solutions, our solution is "autonomous" in the sense that it can dynamically adapt to environment changes and does not need extra scheme to track users' locations. Hence, our work relaxes the requirement of an underlying localization mechanism in existing works.

3.3.2 System Model

3.3.2.1 Light Measurement Method

In our system, there are n users, $u_1, u_2, \ldots, u_n$, m whole lighting devices, D_1, D_2, $\ldots$, D_m, and m' local lighting devices, $d_1, d_2, \ldots, d_{m'}$. The lighting devices are controllable. For all $i = 1, 2, \ldots, n$, user u_i carries a light sensor s_i which periodically reports an illumination level

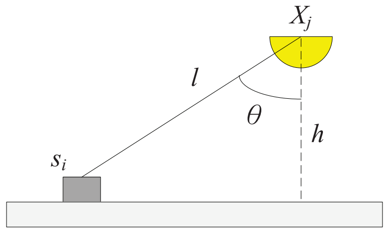
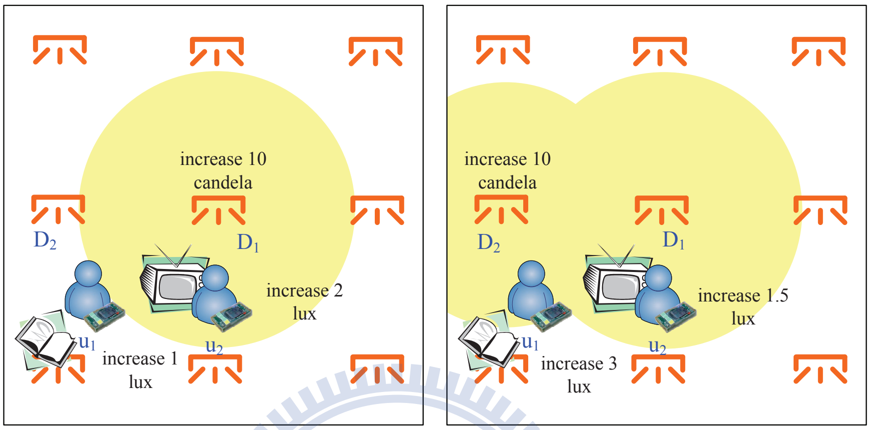


Figure 3.3: Measuring the impact of a light source X_j on a light sensor s_i.



(a) (b)

Figure 3.4: An example of weight measurement.

P_i sensed by the sensor to the control host. The luminous intensity emitted by D_j is denoted by C_j^D, and that by d_j is denoted by C_j^d. Considering physical limitations, we assume that C_j^D and C_j^d should satisfy $C_j^{D_{min}} \leq C_j^D \leq C_j^{D_{max}}$ and $C_j^{d_{min}} \leq C_j^d \leq C_j^{d_{max}}$.

We make the following assumptions in our work. First, there exists a natural light source, but it may change over time. Second, artificial light sources are assumed to be "point-like" ones such as LEDs. This makes modeling the impact of light sources easier. For whole lighting sources, disturbance from other objects may exist (such as furniture, obstacles, walls, etc.). However, we assume that it is possible to derive the impact of a whole lighting device on a sensor. That allows us to decide the proper intensity of each light source. For local lighting sources, we assume that no such disturbance exists. So, we can derive the distance between a lighting source and a user based on the measurement of light sensors. In addition, we assume that local lighting sources are mounted on robot arms and thus the position and orientation of local lighting sources are controllable. We will discuss more about this in Section 3.3.4.

Next, we explain how to model the impact of a light source X_j on a light sensor s_i. Here, X_j can be a whole lighting source D_j or a local lighting source d_j (See Fig. 3.3). Let l and h

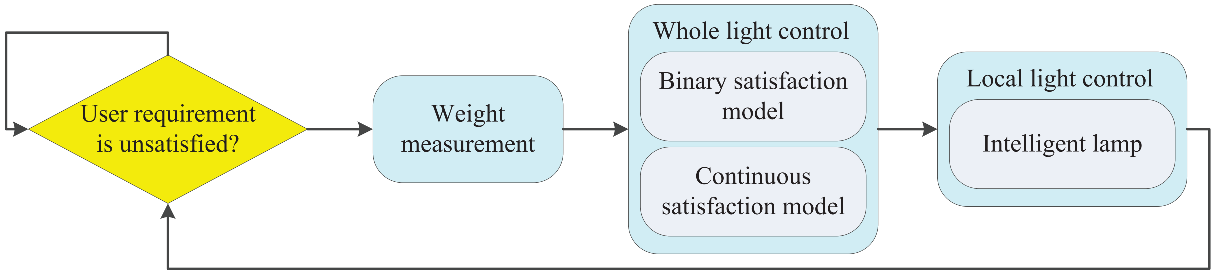


Figure 3.5: Light control flow chart.

be the distances from X_j to s_i and to the nearest ground, respectively. Now let X_j increase its intensity by ΔC_j^X candela, and we measure the change of illumination $\Delta L_{i,j}$ at s_i. According to the light propagation property, we have

$$\Delta L_{i,j} = \frac{\Delta C_j^X \times \cos\theta}{l^2} = \frac{\Delta C_j^X \times h}{l^3}. \tag{3.2}$$

From ΔC_j^X and the observed $\Delta L_{i,j}$, we define the impact of X_j on s_i as

$$w_{i,j}^X = \frac{\Delta L_{i,j}}{\Delta C_j^X} = \frac{h}{l^3}. \tag{3.3}$$

Intuitively, this implies that even if l and h are unknown, we can still measure the weight $w_{i,j}^X$ from ΔC_j^X and $\Delta L_{i,j}$. Therefore, we can easily decide the amount of increment/decrement on X_j's intensity to achieve the desired level of illumination sensed by s_i. Below, if $X_j = D_j$, the impact is written as $w_{i,j}^D$; if $X_j = d_j$, it is written as $w_{i,j}^d$. The measurement of impact values should be done one-by-one, so the overall time complexity is $O(m + m')$. In comparison, this is much lower than those in [45], [46] and [55]. Besides, in our work, we can further consider measuring the impacts of some lighting devices and using interpolation techniques to estimate the impacts of remaining lighting devices to further reduce the measurement cost.

The key technique to the above weight measurement is to slightly increase each light source's intensity one-by-one. In the example illustrated in Fig. 3.4(a), if D_1 increases 10 candela and the light intensities sensed by u_1 and u_2 increase by 1 and 2 lux, respectively, then we can get $w_{1,1}^D = \frac{1}{10} = 0.1$ and $w_{2,1}^D = \frac{2}{10} = 0.2$. Similarly, in Fig. 3.4(b), if D_2 increases 10 candela and the light intensities sensed by u_1 and u_2 increase by 3 and 1.5 lux, then we can get $w_{1,2}^D = \frac{3}{10} = 0.3$ and $w_{2,2}^D = \frac{1.5}{10} = 0.15$.

Because illuminations are additive [55], the light intensity sensed by s_i, denoted as P_i, is the sum of the natural light L_i^{na} and the illuminations provided by the whole and local lighting devices, i.e.

$$P_i \approx \sum_{j=1}^{m} (w_{i,j}^D \times C_j^D) + \sum_{j=1}^{m'} (w_{i,j}^d \times C_j^d) + L_i^{na}. \quad (3.4)$$

P_i is called the concentrated illumination perceived by u_i, and the background illumination perceived by u_i is estimated by $P_i - \sum_{j=1}^{m'} (w_{i,j}^d \times C_j^d)$.

3.3.2.2 Control Flow

Fig. 3.5 illustrates the light control flow of our system. The control process is triggered by user movement, periodical checking, or inputs from sensors which reveal that some users are not satisfied. Then, the weight measurement block determines $w_{i,j}^D$ and $w_{i,j}^d$ that had been discussed in Section 3.3.2.1. After that, the whole light control and local light control modules follow. The background illumination constraints are achieved by tuning C_j^D for all users based on $\sum_{j=1}^{m} (w_{i,j}^D \times C_j^D) + L_i^{na}$, and the concentrate illumination constraints are achieved by tuning C_j^d for each user based on $\sum_{j=1}^{m} (w_{i,j}^D \times C_j^D) + \sum_{j=1}^{m'} (w_{i,j}^d \times C_j^d) + L_i^{na}$. It turns out that decisions of the whole or local light control can be made independently of each other.

3.3.3 Control of Whole Lighting Devices

In this section, we design two solutions for the binary and the continuous satisfaction models. For the binary satisfaction model, the primary goal is to meet all users' requirements. Once the primary goal is met, the secondary goal is to minimize the total energy cost. It is possible that there exists no solution to meet all users' requirements. In such a case, we can compromise by enlarging users' acceptable intervals until all users are satisfied. The details of this issue will be discussed later. After the relaxation, we go to achieve the secondary goal, just like before. For the continuous satisfaction model, since users' satisfaction values are represented by utility functions, there are not too many chances to further optimize the total energy cost. So, in this case, we only adjust light intensities to maximize the total satisfaction value of all users.

3.3.3.1 Binary Satisfaction Model

Our goal is to determine C_j^D for device D_j to meet users' background illumination requirements. Under the binary satisfaction model, we are given the inputs including $w_{i,j}^D$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$; L_i^{na}, R_i^{bl} and R_i^{bu} for all $i = 1, \ldots, n$; $C_j^{D_{min}}$ and $C_j^{D_{max}}$ for all $j = 1, \ldots, m$.

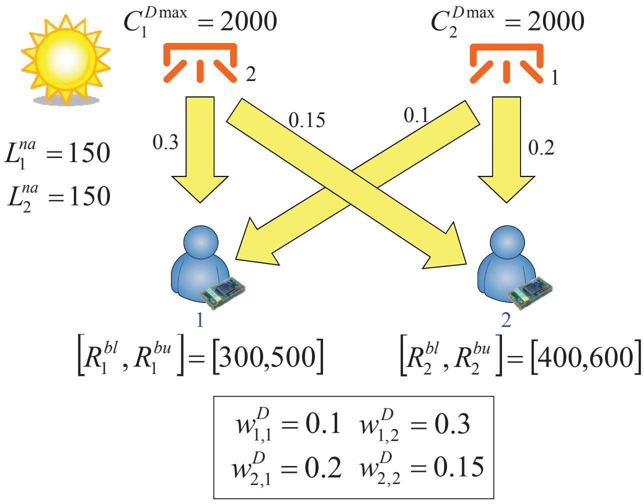


Figure 3.6: An example for the binary satisfaction model.

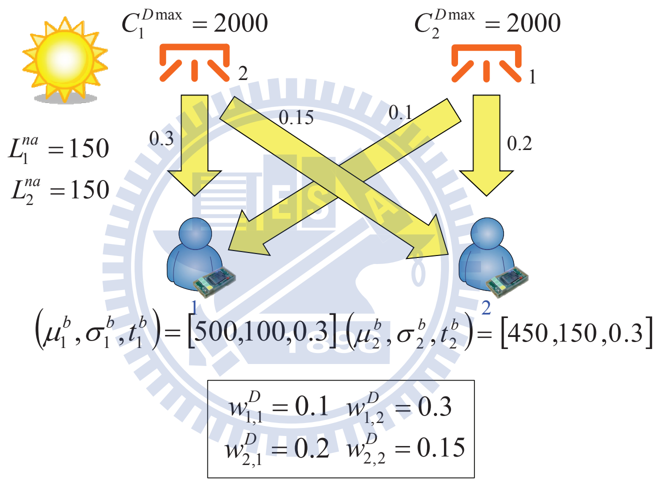


Figure 3.7: An example for the continuous satisfaction model.

Our goal is to solve C_j^D for all $j = 1 \ldots m$ with the objective function:

$$\min \quad \sum_{j=1}^{m} C_j^D \tag{3.5}$$

subject to:

$$R_i^{bl} \leq \sum_{j=1}^{m} (w_{i,j}^D \times C_j^D) + L_i^{na} \leq R_i^{bu} \quad \text{for all} \quad i = 1 \ldots n \tag{3.6}$$

$$C_j^{Dmin} \leq C_j^D \leq C_j^{Dmax} \quad \text{for all} \quad j = 1 \ldots m \tag{3.7}$$

Note that in Eq. (3.6), the $w_{i,j}^D$ can be known by the weight measurement method in Section 3.3.2.1. In some cases, the initial value of C_j^D is not zero. Hence, we can subtract the initial value to get the adjustment value for each lighting device. Eq. (3.5) is to minimize the total power consumption of whole lighting devices. Eq. (3.6) imposes that all users' background illumination requirements should be met. Eq. (3.7) is to confine the adjustment result within the maximum and the minimum bounds. Note that $C_j^{D_{min}} \geq 0$ for all j. This is a linear programming problem and can be solved by, e.g., the Simplex method [13]. Intuitively, the primary goal is to meet all users' requirements. The secondary is to achieve Eq. (3.5). However, in reality, the system may be infeasible. One may try to eliminate the least number of requirements to find a feasible sub-solution. However, it was shown that finding a feasible subsystem of a linear system by eliminating the fewest constraints is NP-hard [49]. So, we compromise by gradually relaxing users' requirements to make this problem feasible. Therefore, we propose an iterative process as follows. First, we run the Simplex method. If no feasible solution is found, we change u_i's requirement to $[\max(0, R_i^{bl} - \alpha), R_i^{bu} + \alpha]$ for each $i = 1 \ldots n$, where α is a constant. Then we run the Simplex method again. This is repeated until a solution is found. Once all users' requirements are met, we go to minimize the total energy cost.

3.3.3.2 Continuous Satisfaction Model

Under the continuous satisfaction model, the inputs include $w_{i,j}^D$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$; $C_j^{D_{min}}$ and $D_j^{D_{max}}$ for all $j = 1, \ldots, m$; $(\mu_i^b, \sigma_i^b, t_i^b)$ and L_i^{na} for all $i = 1, \ldots, n$. The goal is to find C_j^D for all $j = 1 \ldots m$ with the objective function:

$$\max \quad \sum_{i=1}^{n} f_i^b\left(\sum_{j=1}^{m} (w_{i,j}^D \times C_j^D) + L_i^{na}\right) \tag{3.8}$$

subject to:

$$\mu_i^b - \sigma_i^b\sqrt{-2ln(t_i^b)} \leq \sum_{j=1}^{m} (w_{i,j}^D \times C_j^D) + L_i^{na} \leq \mu_i^b + \sigma_i^b\sqrt{-2ln(t_i^b)} \quad \text{for all} \quad i = 1 \ldots n \tag{3.9}$$

$$C_j^{D_{min}} \leq C_j^D \leq C_j^{D_{max}} \quad \text{for all} \quad j = 1 \ldots m \tag{3.10}$$

Eq. (3.8) is to maximize the sum of satisfaction values of all users. Eq. (3.9) imposes that all users' background illumination requirements should be met. Eq. (3.10) specifies the bounds. This is a non-linear programming problem and can be solved by a sequential quadratic programming (SQP) method [9]. If there is no feasible solution, we gradually relax users' requirements to make this problem feasible. We propose an iterative process as follows: First, we run the SQP method. If no feasible solution is found, we change u_i's background threshold

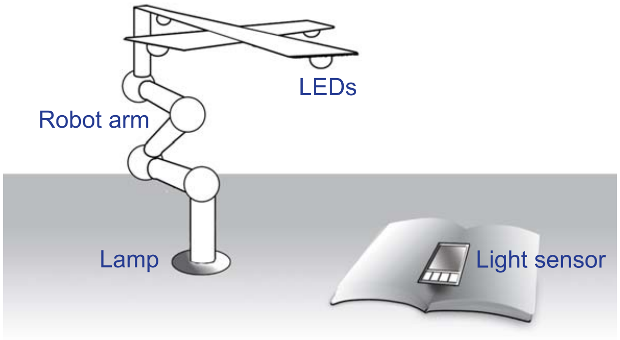


Figure 3.8: Service scenario of an iLamp and a light sensor.

to $\max(0, t_i^b - \beta)$ for each $i = 1 \ldots n$, where β is a constant. Then we run the SQP method again. This is repeated until a solution is found.

3.3.3.3 Examples

An example of the binary satisfactory model is illustrated in Fig. 3.6 where there are two users u_1 and u_2, and two whole lighting devices D_1 and D_2. We have $L_1^{na} = 150$, $L_2^{na} = 150$, $[R_1^{bl}, R_1^{bu}] = [300, 500]$, and $[R_2^{bl}, R_2^{bu}] = [400, 600]$. The objective function is

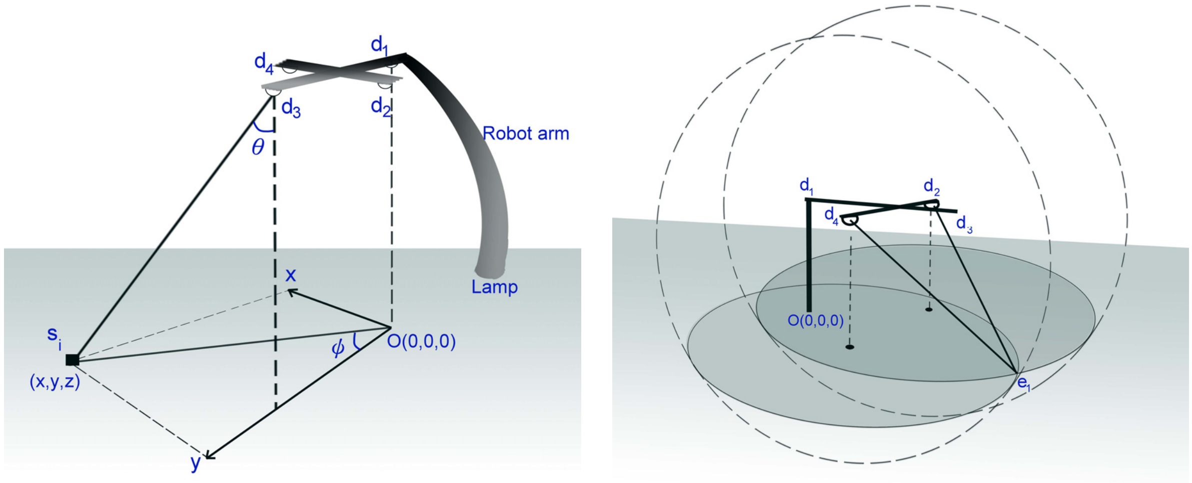
$$\min \quad C_1^D + C_2^D$$

subject to :

$$\begin{array}{rcccl} 300 & \leq & 150 + 0.1 \times C_1^D + 0.3 \times C_2^D & \leq & 500 \\ 400 & \leq & 150 + 0.2 \times C_1^D + 0.15 \times C_2^D & \leq & 600 \\ 0 & \leq & C_1^D & \leq & 2000 \\ 0 & \leq & C_2^D & \leq & 2000 \end{array}$$

Because this problem is feasible, the solution is $C_1^D = 1166.67$ and $C_2^D = 111.11$. If the current light intensities are $C_1^D = 100$ and $C_2^D = 100$, we need to set $\Delta C_1^D = 1166.67 - 100 = 1066.67$ and $\Delta C_2^D = 111.11 - 100 = 11.11$.

Fig. 3.7 illustrates an example of the continuous satisfaction model. Similarly, there are two users and two lighting devices. The natural lights are $L_1^{na} = 150$ and $L_2^{na} = 150$. We assume that $(\mu_1^b, \sigma_1^b, t_1^b) = (500, 100, 0.3)$ and $(\mu_2^b, \sigma_2^b, t_2^b) = (450, 150, 0.3)$ for u_1 and u_2, respectively. Given $t_1^b = 0.3$ and $t_2^b = 0.3$, we can derive $[\mu_1^b - \sigma_1^b\sqrt{-2ln(t_1^b)}, \mu_1^b + \sigma_1^b\sqrt{-2ln(t_1^b)}] =$



(a) (b)

Figure 3.9: The geometry model of iLamp to track the location of s_i.

$[345, 655]$ and $[\mu_2^b - \sigma_2^b\sqrt{-2ln(t_2^b)}, \mu_2^b + \sigma_2^b\sqrt{-2ln(t_2^b)}] = [167, 633]$ for u_1 and u_2, respectively. The objective function is:

$$\begin{aligned} \max \quad & f_1^b(150 + 0.1 \times C_1^D + 0.3 \times C_2^D) + \\ & f_2^b(150 + 0.2 \times C_1^D + 0.15 \times C_2^D) \end{aligned}$$

subject to:

$$\begin{array}{rcccl} 345 & \leq & 150 + 0.1 \times C_1^D + 0.3 \times C_1^D & \leq & 655 \\ 167 & \leq & 150 + 0.2 \times C_2^D + 0.15 \times C_2^D & \leq & 633 \\ 0 & \leq & C_1^D & \leq & 2000 \\ 0 & \leq & C_2^D & \leq & 2000 \end{array}$$

Again, this problem is also feasible. The solution is $C_1^D = 355.4$ and $C_2^D = 1066.3$. If the current light intensities are $C_1^D = 100$ and $C_2^D = 100$, we need to set $\Delta C_1^D = 355.4 - 100 = 255.4$ and $\Delta C_2^D = 1066.3 - 100 = 966.3$.

3.3.4 Control of Local Lighting Devices

The above lighting control heuristic algorithms are able to adjust background illuminations to meet users' needs. Here, we are further going to propose a robotic device, called *Intelligent Lamp* (iLamp), to provide concentrated illuminations. An iLamp has a robot arm with at least four local lighting devices and is supposed to serve one user who has need of concentrated illumination. The service scenario is shown in Fig. 3.8. The sensor should be placed on the reading surface. On detecting a user under its service area, the iLamp will compute its relative location to the light sensor, move via its robot arm to a better location, and then adjust its luminous intensities to meet the need with the least energy. Detecting a nearby user is a simple job since a local lighting device can check if it has non-negative impact on a sensor.

Given an iLamp and a light sensor s_i, they will cooperate with each other by the following four steps to achieve our goal: (1) collect the current P_i sensed by s_i, (2) calculate the location of s_i, (3) adjust the lamp's robot arm, and (4) adjust the luminous intensities of its lighting devices. Step 1 is executed periodically. Once it finds that the current illumination falls outside the required interval, steps 2, 3, and 4 are triggered. Central to our scheme is step 2, so we will elaborate it in more details below.

To drive step 2, assume for simplicity that the iLamp has four local lighting devices d_1, d_2, d_3, and d_4 as shown in the geometry model in Fig. 3.9(a). Note that it is not hard to extend this result to more lighting devices in other geometry models. Since there is a robot arm, the iLamp should know the coordinate (x_j, y_j, z_j) of d_j, $j = 1 \dots 4$. Without loss of generality, regard the projection of d_1 on the reading surface as the origin $O(0, 0, 0)$, the projection of $\overrightarrow{d_1 d_3}$ on the surface as the y axis, the projection of $\overrightarrow{d_2 d_4}$ on the surface parallel with the x axis, and the norm of the surface toward the sky as the z axis. Let the location of s_i be $(x, y, z = 0)$. We will derive a scheme to find its location as follows. Since LED is a point-like light source, it dissipates identically in all directions. Our scheme consists of two symmetric processes. The first one is to use d_2 and d_4 to estimate two potential locations of s_i and then use d_1 and d_3 to screen out one location. The second one is to use d_1 and d_3 to estimate two potential locations of s_i, and use d_2 and d_4 to screen out one location. Finally, we take their middle point as the estimated location of s_i.

1. For each d_j, $j = 1 \dots 4$, increase its luminous intensity by ΔC_j candela and measure the change of illuminance intensity at s_i, denote by ΔL_{ij}. According to the definition of illumination, we have the equality:

$$\Delta L_{ij} = \frac{\Delta C_j \times \cos\theta_{ij}}{(\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2})^2},$$

where

$$\cos\theta_{ij} = \frac{z_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}}.$$

This leads to

$$\Delta L_{ij} = \frac{\Delta C_j \times z_j}{(\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2})^3}. \tag{3.11}$$

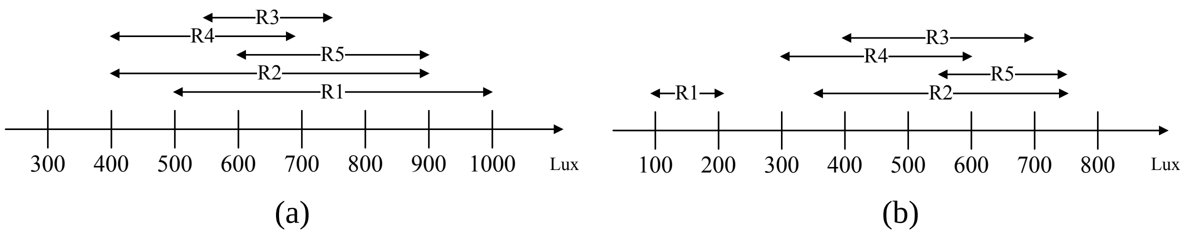


Figure 3.10: Requirement pools: (a) $RP1$ and (b) $RP2$.

2. Observe that the equations of ΔL_{i2} and ΔL_{i4} represent two balls centered at d_2 and d_4, respectively. Since it is known that $z = 0$, each of these two balls intersects with plane $z = 0$ at a circle. These two circles will intersect at two points. Using any equation of ΔL_{i1} and ΔL_{i3}, we can pick one point as the estimated location of s_i, called e_1. (Refer to Fig. 3.9(b).)

3. Similarly, the equations of ΔL_{i1} and ΔL_{i3} represent two balls at d_1 and d_3, respectively, each intersecting with plane $z = 0$ at a circle. Again, these two circles intersect at two points, and we can pick one point as the location of s_i, called e_2, with the assistance of ΔL_{i2} and ΔL_{i4}.

4. Finally, the location of s_i is predicted as the middle point of e_1 and e_2.

In step 3, we move our lighting devices toward the upper side of s_i. This includes two sub-steps. First, we rotate the robot arm by ϕ angle such that the vector from d_1 to d_3, after projecting to the reading surface, is pointing toward the location of s_i. Second, it moves to the upper side of s_i to provide a proper reading angle (a typical angle is $60°$).

Step 4 is to adjust C_j^d, $j = 1 \ldots 4$ to meet the concentrated illumination demand of u_i. From the results in Section 3.3.3, some background and natural illuminations have already been provided. So we only need to add some more light to meet u_i's need. The results in Section 3.3.3 can be directly applied again here, so we omit the details.

3.3.5 Simulation Results

To understand how our schemes for whole lighting control meet users' requirements while saving energy, we have developed a simulation. Two scenarios are considered. One scenario, denoted as $S1$, is in a room of size 10×10 m^2 with an array of 5×5 whole lighting devices. The other scenario, denoted as $S2$, is in a room of size 20×20 m^2 with an array of 9×9 while lighting devices. We set $C_i^{D_{min}} = 0$ and $C_i^{D_{max}} = 3000$ for all i. Our proposed algorithms are compared to other two schemes, called FIX and GREEDY. The FIX scheme is a very intuitive one assuming that users' locations are known in advance. The nearest devices are selected and set to a fixed candela value n. We denote this scheme as FIX-n below. The GREEDY scheme also assumes that users' locations are known. For each user, the nearest device is selected to satisfy the user (if possible). If it is still below the required illumination, the second nearest device is picked to increase the intensity. This is repeated until the user is satisfied. Note that it may happen that a user is satisfied first but later on becomes unsatisfied due to other devices change their intensities. Below, we compare the outcomes according the two satisfaction models introduced in Section 3.3.2.

- Binary satisfaction model: A requirement pool is a set of requirements. In the simulation, each user will randomly choose one from the pool as its requirement. Two requirement pools, denoted by $RP1$ and $RP2$, are considered. See Fig. 3.10 in which each range R_i represents an expected illumination interval. We consider two performance indices here. The first index is the total energy consumption. The second index is called GAP, which reflects the difference between the provided illumination and the required one. The GAP for user u_i is

$$GAP(u_i) = \begin{cases} 0 & \text{if } R_i^{bl} \leq P_i \leq R_i^{bu} \\ \min(|R_i^{bl} - P_i|, |R_i^{bu} - P_i|) & \text{otherwise.} \end{cases} \tag{3.12}$$

  We will measure the average GAP of all users.

  Fig. 3.11 shows our simulation results under different combinations of $S1/S2$ and $RP1/RP2$. In Fig. 3.11(a), we see that our scheme is the most energy-efficient while keeps the average GAP close to zero. This is because the requirement intervals in $RP1$ have common overlapping and that allows our system to satisfy all users in most cases. Note that although FIX-1000 uses less energy, its GAP is much larger. Fig. 3.11(b) adopts $RP2$. Because some requirements are violated, our scheme also induces some GAP. However, our scheme is also the most energy-efficient. Fig. 3.11(c) and Fig. 3.11(d) are the case of $S2$ and the trends are similar. This demonstrates that our scheme is quite scalable to network size.

- Continuous satisfaction model: We define two requirement pools $RP3$ and $RP4$, as shown in Fig. 3.12. Note that $RP4$ has higher deviation in requirements than $RP3$.

The satisfaction threshold t is set to 0.3. We compare two performance indices: average user satisfaction and energy consumption. Fig. 3.13 shows our simulation results under different combinations of $S1/S2$ and $RP3/RP4$. These results consistently indicate that our scheme provides the highest satisfaction levels and outperforms FIX and GREEDY schemes in energy consumption.

3.3.6 Prototyping Results

We have developed a prototype to verify our designs. Fig. 3.14 shows the system architecture. Users carry a badge with a built-in light sensor. The user's preference can be configured via the badge. The control host can make decisions and send them to lighting devices. We test our system in a room of size $8 \times 6\ m^2$ with whole lighting devices deployed in a grid manner. Some demo videos of our work can be found at `http://hscc.cs.nctu.edu.tw/iLight/`. Below, we introduce each device, and then give our testing results.

3.3.6.1 User Badge and Light Sensor

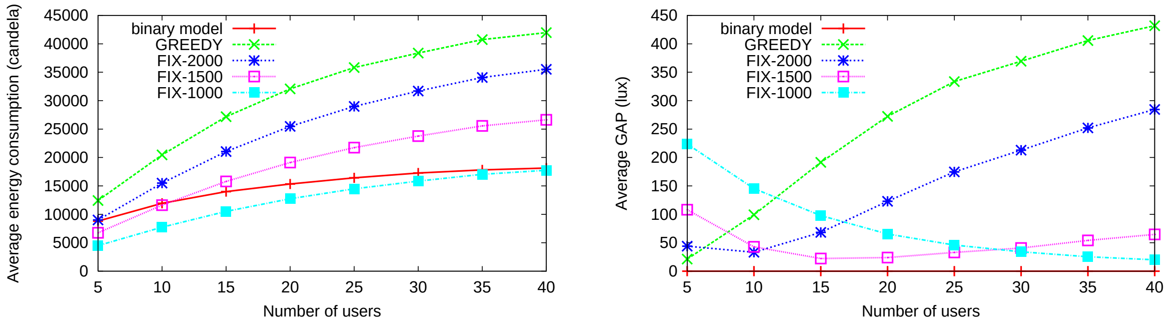
The user badge has a wireless module Jennic (JN5139) [26], a light sensor (TSL230) [35], a TFT LCD panel (ILI9221) [60], and some input buttons. JN5139 is a single-chip microprocessor with an IEEE 802.15.4 [23] module. Fig. 3.15(a) illustrates the front and back views of the user badge. The outlook of a badge is like a bookmark. Users can specify their preferences via our graphic user interface (GUI) and buttons.

3.3.6.2 Whole Lighting Device

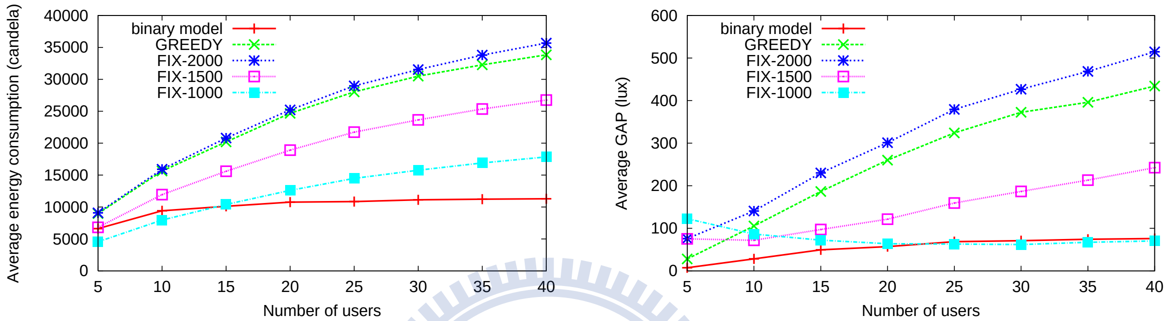
We use LEDs as light sources and deploy them on the ceiling. As shown in Fig. 3.15(b), each whole lighting device has a 4×4 LED module and a thermal pad is attached on its back for heat dissipation. We adopt pulse width modulation of digital input/output (DIO) to control the luminous intensity of light sources. Each LED has 20 levels of illumination, ranging from 0% to 100% luminous intensity. Fig. 3.15(c) shows our prototype.

3.3.6.3 iLamp

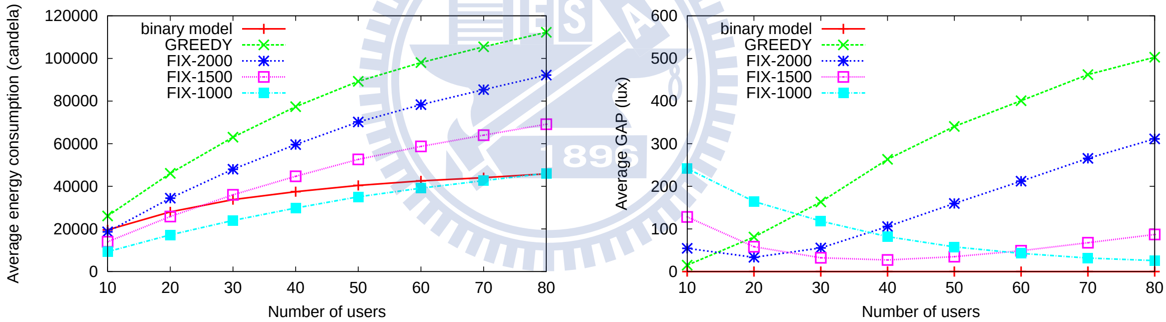
Fig. 3.15(d) shows the iLamp, which is composed of a robot arm, four sets of LEDs, and a JN5139 module. The robot arm consists of six Dynamical AX-12 actuators [5] as the lamp



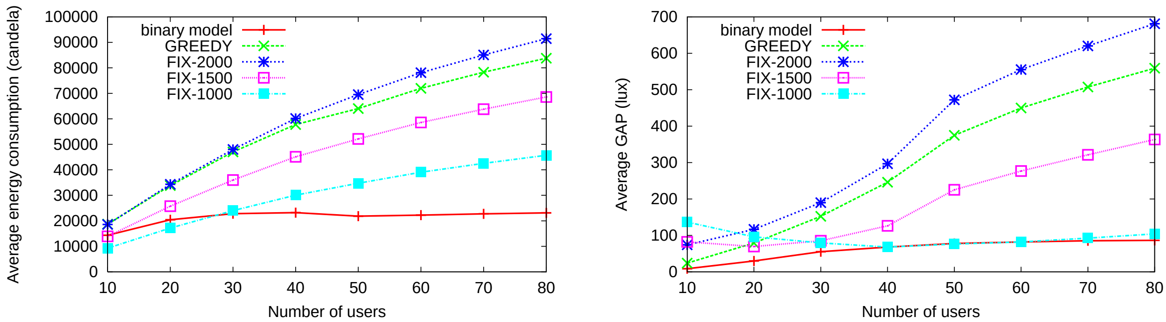
(a)



(b)

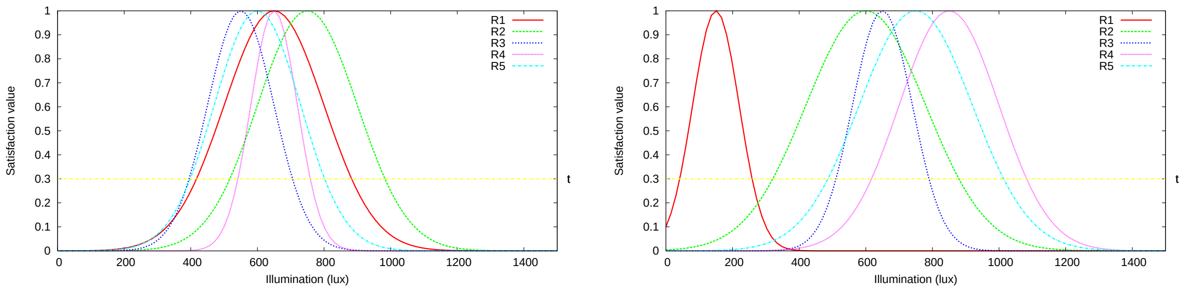


(c)



(d)

Figure 3.11: Comparison under the binary satisfaction model: (a) network scenario $S1$ and pool $RP1$, (b) network scenario $S1$ and pool $RP2$, (c) network scenario $S2$ and pool $RP1$, and (d) network scenario $S2$ and pool $RP2$.



(a) (b)

Figure 3.12: Requirement pools: (a) $RP3$ and (b) $RP4$.

holder. Each AX-12 actuator can rotate from $0°$ to $300°$ at an accuracy of $0.33°$. LEDs are similar to those used in whole lighting devices and with 20 levels of illuminations.

3.3.6.4 Control Host

Implemented by JAVA, the control host is the core of our system. It is composed of three components, including the *User Status Tracker*, *Decision Handler*, and *Device Controller*. By applying Java thread programming techniques, tasks are handled concurrently.

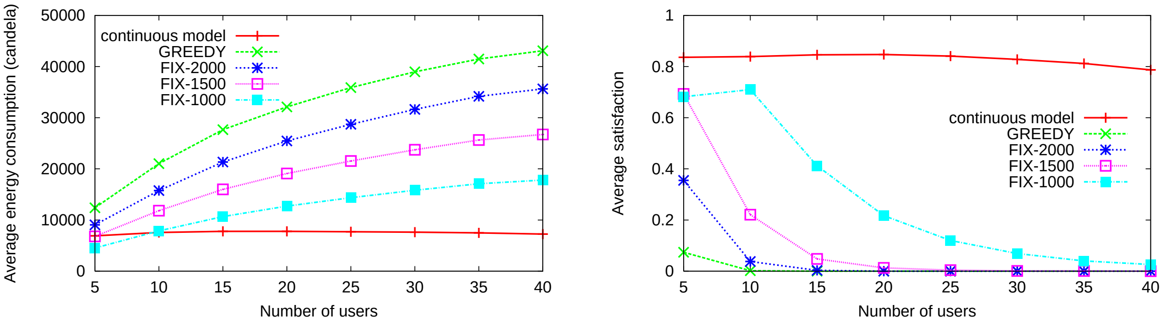
- User Status Tracker: This component checks current illuminations of all users periodically and, when needed, updates users' requirements. If it finds that a user's requirement is not satisfied or is updated, the Decision Handler will be triggered.

- Decision Handler: This component realizes our control algorithms. It is triggered by the User Status Tracker. The linear and non-linear programming are resolved by the MATLAB Builder for Java [42]. The results are sent to the Device Controller to adjust lighting devices

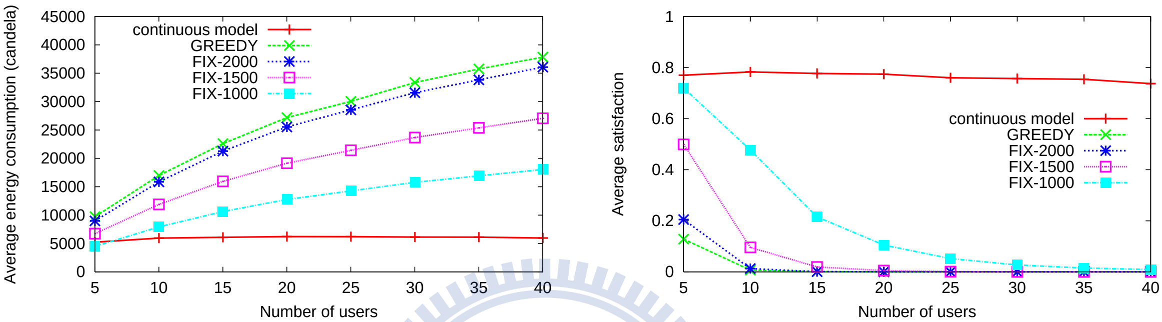
- Device Controller: This is the interface between the control host and actuators. Commands are sent via RS232.

3.3.6.5 Performance Verification

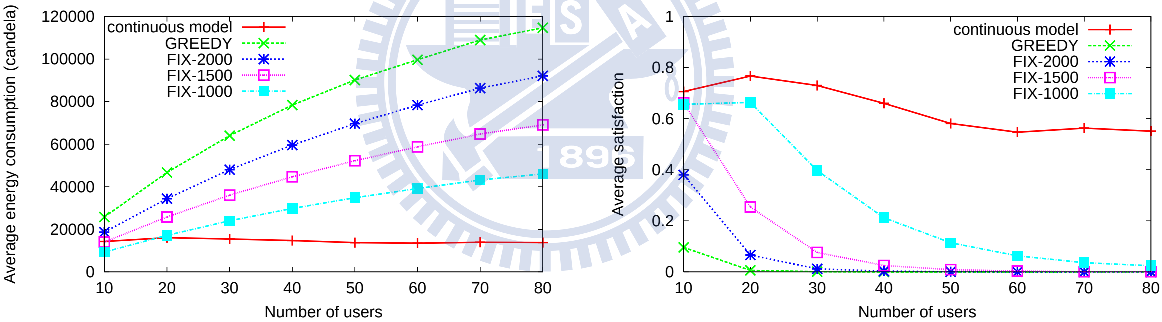
We first verify the correctness of Eq. (3.2) through some tests. In Fig. 3.16(a), we fix h and ΔC_j^X and vary l from 30 to 230 cm. We measure the received illumination $\Delta L_{i,j}$ (in dash lines) and derive the ideal illumination (in solid line). The distance error between ideal and real values are calculated (in bars). As can be seen, the distance error is bounded by 10 cm. In Fig. 3.16(b),



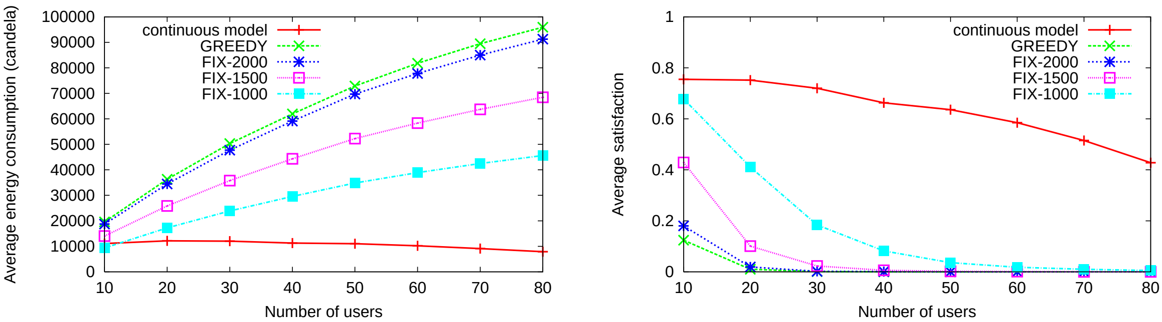
(a)



(b)

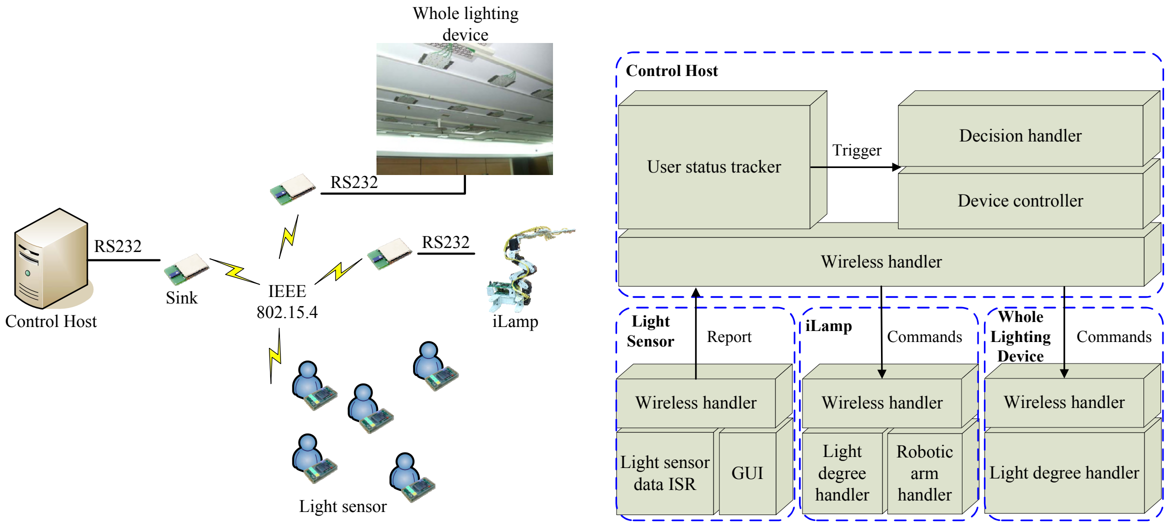


(c)



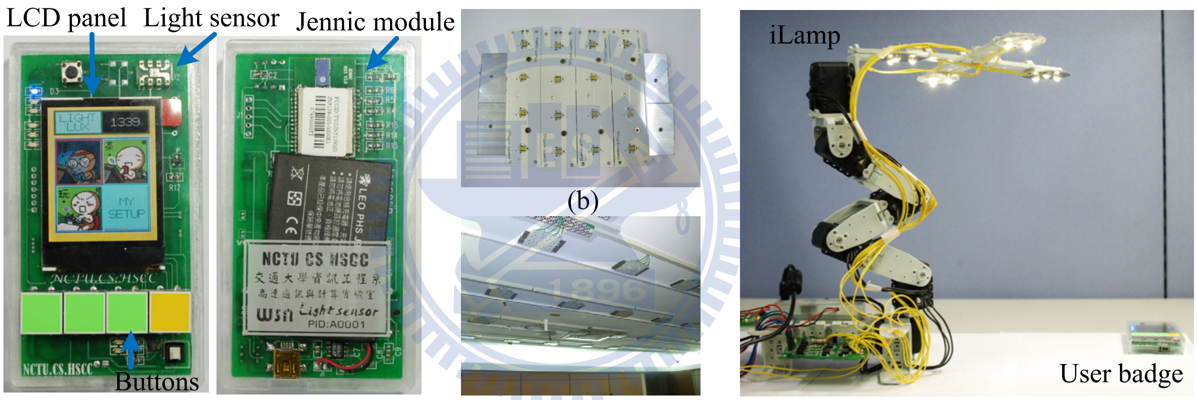
(d)

Figure 3.13: Comparison under the continuous satisfaction model: (a) network scenario $S1$ and pool $RP3$, (b) network scenario $S1$ and pool $RP4$, (c) network scenario $S2$ and pool $RP3$, and (d) network scenario $S2$ and pool $RP4$.



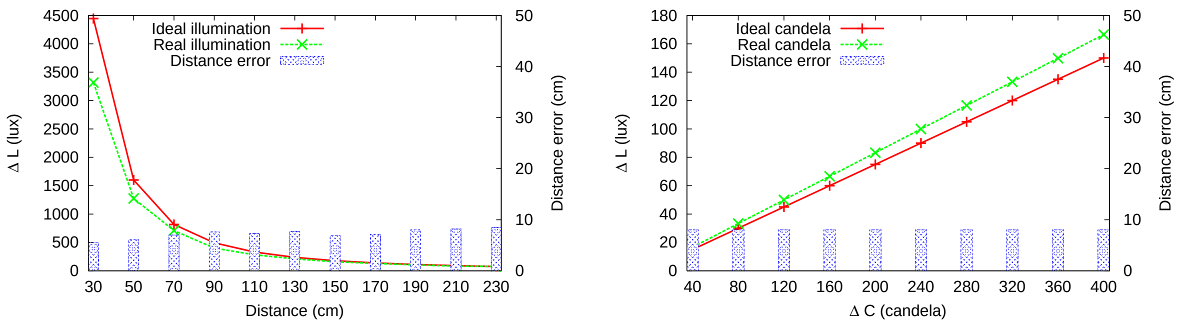
(a) (b)

Figure 3.14: Hardware and software system architecture of our prototype.



(a) (c) (d)

Figure 3.15: (a) Front and back views of user badge, which looks like a bookmark, (b) whole lighting device, (c) testing environment, and (d) iLamp.



(a) (b)

Figure 3.16: Verification of Eq. (3.2): (a) fixing h and ΔC_j^X and varying l. (b) fixing h and l and varying ΔC_j^X.

we fix h and l and vary ΔC_j^X from 40 to 400. We measure $\Delta L_{i,j}$ and calculate the distance error. Again, the gaps between the real distance and the derived distance are quite small.

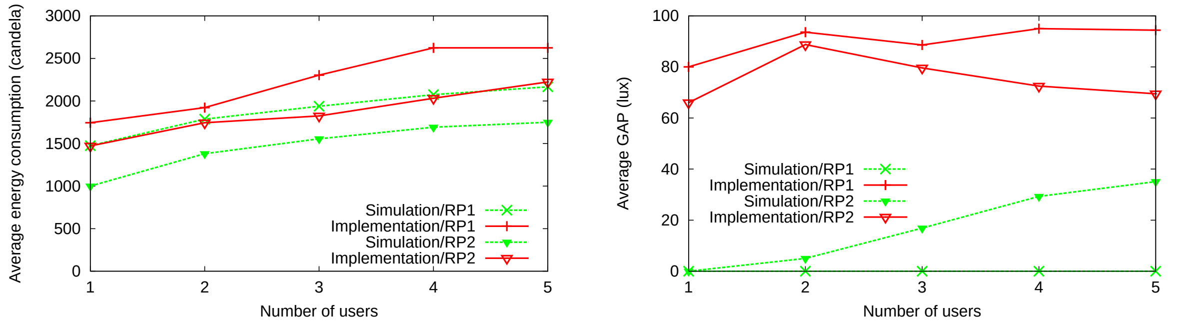
Next, we verify our results in a real environment. We place 20 whole lighting devices in a 8×6 m^2 room in a grid-like manner. We adopt the same performance indices introduced in Section 3.3.5. In Fig. 3.17, we compare the simulation and implementation results where there are 1 to 5 users. Each user may randomly select a requirement from requirement pools $RP1$ and $RP2$ for the binary satisfaction model or $RP3$ and $RP4$ for continuous satisfaction model. In all performance indices (average energy consumption, GAP, and satisfaction), the difference between simulation and implementation results are very close. The results indicate the correctness of our approaches. We further validate our results under different weather conditions, which may reflect to different nature light scenarios. Fig. 3.18 shows the results by using the measured natural light as the index. We fix the number of users to three and randomly select their lighting requirements. There still exists high consistency between simulation and implementation results in most cases. Also, when the average natural light increases, the gap in energy consumption tends to become smaller.

For iLamp, we place a book on the desk and attach the light sensor on the book. We try to move the book (i.e, move the light sensor) from time to time. In this experiments, the ΔC is set to 12.25 candela. The average time for executing the algorithm of iLamp are about 3 seconds. (The steps 1 and 2 take about one second, and steps 3 and 4 take about two seconds.) This experiment shows that the amount of error of the angle pointing toward the light sensor is less than 15°.

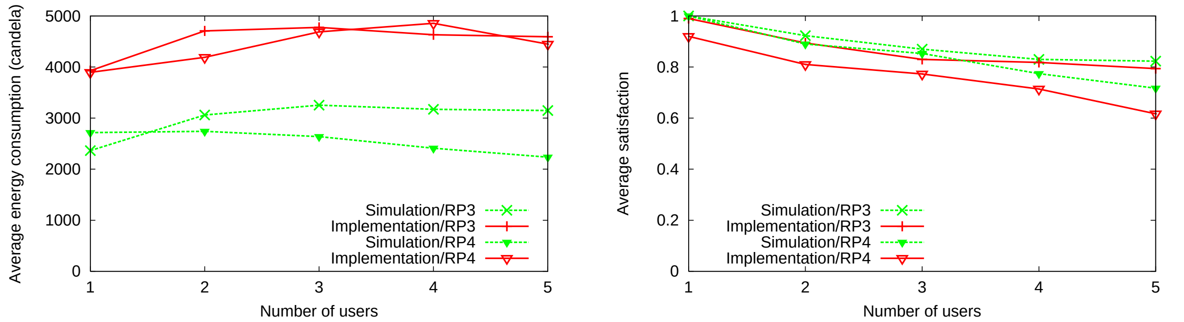
3.4 An Intelligent Air Conditioners Control System for a Smart Living Space

3.4.1 Motivation

According to the report in [18], more than one third of electricity is spent on HVAC (Heating, Ventilating, and Air Conditioning) systems. Hence, how to adjust the HVAC system to achieve energy saving and satisfy users' requirements is an important issue. Traditionally, PID (Proportional - Integral - Derivative) controller [48] is a common control mechanism. PID controller, which attempt to correct the error between the measured and desired value by a feedback mechanism, is the generic control loop feedback mechanism. PID controller can quickly converge

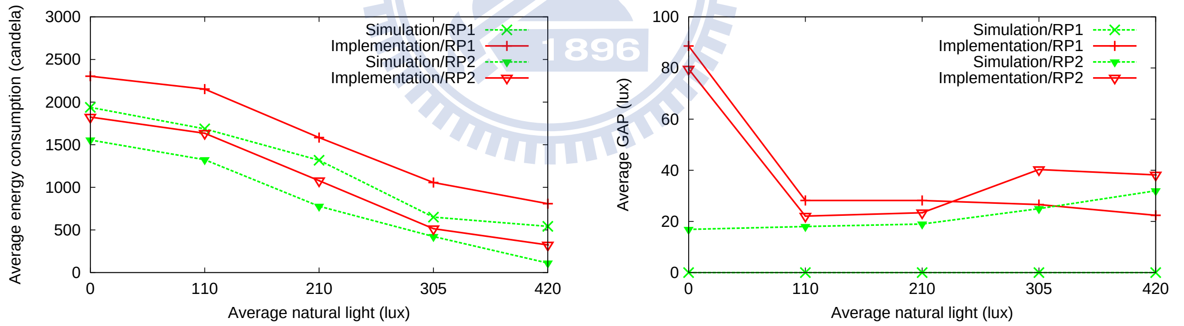


(a)

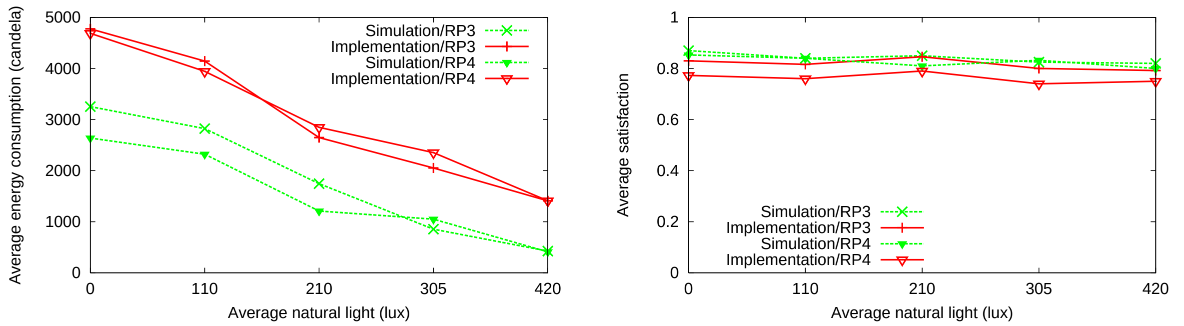


(b)

Figure 3.17: Comparison of simulation and real implementation results under different numbers of users: (a) binary satisfaction model and pool $RP1$ or $RP2$, and (b) continuous satisfaction model and pool $RP3$ or $RP4$.



(a)



(b)

Figure 3.18: Comparison of simulation and real implementation results under different natural light: (a) binary satisfaction model and pool $RP1$ or $RP2$, and (b) continuous satisfaction model and pool $RP3$ or $RP4$.

while setting parameters suitably. However, the PID controller is not suitable for air conditioners control because temperature needs long time to achieve the steady state. For example, when we turn on the air conditioner, indoor temperature needs a period of time to achieve the stable state. Recent year, many control strategies for HVAC have been proposed, such as neural network based [33], fuzzy logic based [2], and wireless sensor network based control system [37], etc. The authors [33] present a thermal comfort controller for HVAC application. The system can maintain the indoor comfort level within the desired range under both heating or cooling modes. The work [2] presents a fuzzy logic controllers to control HVAC system concerning energy consumption and indoor comfort requirements. The work [37] propose a multi-sensor HVAC control system. Through the result of multiple sensors, the system try to minimize discomfort and energy consumption in the environment. The above systems [2][33][37] do not consider users' temperature requirements or each user may need different temperature requirement. Hence, in this section, we propose a WSN-based air conditioner controlling system considering users' temperature requirements. We propose a temperature model and control mechanisms that can adjust the air conditioner quickly to satisfy all users' requirements.

3.4.2 System Model

There are q fixed sensors, m air conditioners, and n users in this environment. The fixed sensor s_k, $k = 1 \dots q$, are randomly deployed in the environment and can report current temperature v_k to the sink node. The air conditioners can be controlled by several levels and the higher number of levels mean the air conditioners need more energy consumption. The air conditioners and their levels are denoted as a_j and d_j, $j = 1 \dots m$, respectively. Considering physical limitations, we assume that d_j should satisfy $d_j^{min} \leq d_j \leq d_j^{max}$. Each user u_i, $i = 1 \dots n$, carries a portable wireless sensor s_{u_i}. Portable wireless sensors will report users' current temperature v_{u_i} to the control server. We also assume that via a localization scheme (such as [31]), users' current locations are known to the control host.

In our system, fixed and portable sensors periodically report their readings to the sink node and the control server can control the air conditioners according their readings. Therefore, our goal is to adjust the air conditioners to satisfy the most of users in this environment and achieve energy saving.

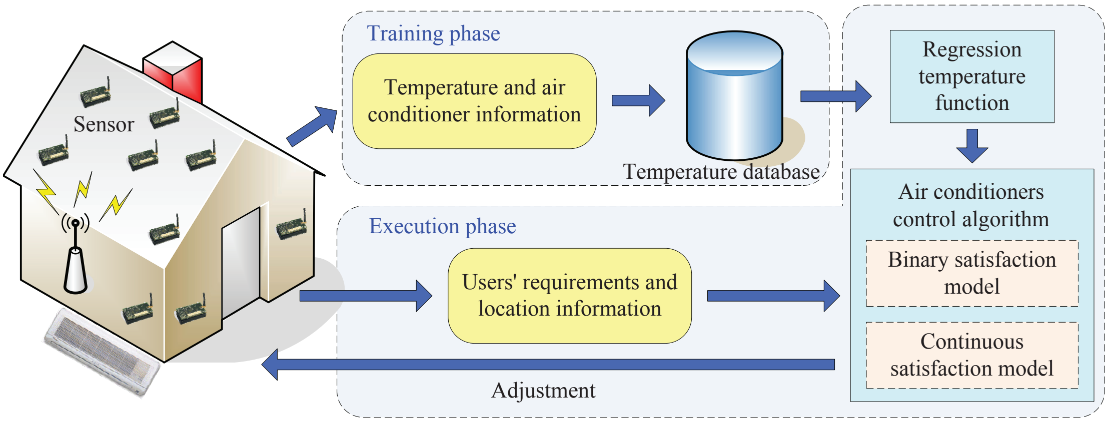


Figure 3.19: The system flow of air conditioners controlling system

3.4.3 Air Conditioner Controlling Scheme

In this section, we propose the air conditioner controlling schemes to satisfy the most of users and achieve energy saving. For two users' requirement models in Section 3.2, we design two solutions for the binary and continuous satisfaction models, respectively. Fig. 3.19 shows the system flow of our air conditioner controlling scheme. The controlling scheme can be divided into two phases. The first one is the training phase. Before running our algorithm, we should collect some training data in training phase. The second one is the execution phase. In execution phase, according to the values which collected in phase one, we can construct the regression temperature model. Then the control server can do a suitable decision for minimizing energy consumption or maximizing users' satisfactions in binary satisfaction or continuous satisfaction model, respectively.

3.4.3.1 Training Phase

In this phase, our system adjusts air conditioners to a set of control levels. After a time period T, we collects the corresponding temperature values from fixed sensors. Here, we assume that the temperature in the indoor environment can be stable after the time period T. We insert the set of control levels and sensor readings into database. Repeat above steps l time, where $l > m+1$, then we can get a number of training data in the database. Therefore, from this training phase, we can get a set of control levels $(d_1^{(l)}, d_2^{(l)}, \ldots, d_m^{(l)})$ of m air conditioners and corresponding temperature values $(v_1^{(l)}, v_2^{(l)}, \ldots, v_q^{(l)})$ of q fixed senors. Note that these training data also can be got by previous execution results.

3.4.3.2 Execution Phase

The execution phase are divided into two steps. First, we construct the regression temperature models for the fixed sensors which locations are nearby users. According to these regression functions, the air conditioners control algorithm can decide a set of control levels of air conditioners for binary or continuous satisfaction models.

Regression temperature function: From above training phase, we can get a set of control levels $(d_1^{(l)}, d_2^{(l)}, \ldots, d_m^{(l)})$ of m air conditioners and corresponding temperature values $(v_1^{(l)}$, $v_2^{(l)}$, ..., $v_q^{(l)})$ of q fixed senors. For each fixed sensor s_k, $k = 1 \ldots q$, based on linear regression, we can assume the following linear relation F_{s_k}:

$$F_{s_k}(d_1, d_2, \ldots, d_m) = \sum_{j=1}^{m} d_j^{(l)} \times b_{j,k} + b_{0,k} = v_k^{(l)}. \tag{3.13}$$

According to Eq. (3.13) and training data, we can construct following matrix:

$$\underbrace{\begin{bmatrix} 1 & d_1^{(1)} & d_2^{(1)} & \cdots & d_m^{(1)} \\ 1 & d_1^{(2)} & d_2^{(2)} & \cdots & d_m^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_1^{(l)} & d_2^{(l)} & \cdots & d_m^{(l)} \end{bmatrix}}_{D} \times \underbrace{\begin{bmatrix} b_{0,k} \\ b_{1,k} \\ \vdots \\ b_{m,k} \end{bmatrix}}_{B_k} = \underbrace{\begin{bmatrix} v_k^{(1)} \\ v_k^{(2)} \\ \vdots \\ v_k^{(l)} \end{bmatrix}}_{V_k}. \tag{3.14}$$

Then the value of matrix B_k can be measured by the least-square analysis:

$$B_k = (D^T D)^{-1} D^T V_k. \tag{3.15}$$

Then, we can get the regression function $F_{s_k}(d_1, d_2, \ldots, d_m)$ for fixed sensor s_k, where $k = 1 \ldots q$.

Air conditioners control algorithm: In this step, we consider binary and continuous satisfaction model to minimize energy consumption and maximize users' satisfactions, respectively.

1. *Binary satisfaction model*: Here, we want to find the optimal control levels $(d_1, d_2, \ldots, d_m)$ for those regression functions F_{s_k} which locations are nearby users. Our goal is to adjust the air conditioners to satisfy the users' temperature requirements and minimize energy consumption.

   Under the binary satisfaction model, we are given the input including R_i^{tl} and R_i^{tu} for all $i = 1 \ldots n$; d_j^{min} and d_j^{max} for all $j = 1 \ldots m$; $F_{s_k}(d_1, d_2, \ldots, d_m)$ for all $k = 1 \ldots q$. Our goal is to solve d_j, for all $j = 1 \ldots m$ with the objective function:

$$\min \quad \sum_{j=1}^{m} d_j \tag{3.16}$$

subject to:

$$R_i^{tl} \le \overline{F_{u_i}}(d_1, d_2, \ldots, d_m) \le R_i^{tu}, \quad \text{for all} \quad i = 1 \ldots n \tag{3.17}$$

$$d_j^{min} \le d_j \le d_j^{max}, \quad \text{for all} \quad j = 1 \ldots m \tag{3.18}$$

Note that in Eq. (3.17), $\overline{F_{u_i}}(d_1, d_2, \ldots, d_m) = \sum_{\forall s_k \in \mathbb{S}} F_{s_k}(d_1, d_2, \ldots, d_m)/|\mathbb{S}|$, where $\mathbb{S} = \{s_k | \forall k, dist(s_k, u_i) < \delta\}$, $dist(A, B)$ is the distance between object A and B, and δ is an adjustable constant. It means we choose several fixed sensor nodes s_k near the user u_i to stand for the temperature that user is located on. Eq. (3.16) is to minimize the total energy consumption of all air conditioners. Eq. (3.17) imposes that all users' temperature requirements should be met. Eq. (3.18) is to confine the control level within the maximum and the minimum bounds. This is a linear programming problem and can be solved by e.g., the Simplex method [13]. In reality, the system may be infeasible. One may try to eliminate the least number of requirements to find a feasible sub-solution. However, it was shown that finding a feasible subsystem of a linear system by eliminating the fewest constraints is NP-Hard [49]. So, we compromise by gradually relaxing users' requirements to make this problem feasible. Therefore, we propose an iterative process as follows. First, we run the Simplex method. If no feasible solution is found, we change u_i's temperature requirement to $[R_i^{tl} - \alpha, R_i^{tu} + \alpha]$ for each $i = 1 \ldots n$, where α is a constant. Then we run the Simple method again. This is repeated until a solution is found or the retry limits we meet. The results of control levels and corresponding sensor readings should be inserted into database to be an new entry.

2. *Continuous satisfaction model*: Here, we want to find the optimal control levels $(d_1$, d_2, ..., $d_m)$ for those regression functions F_{s_k} which locations are nearby users. Our goal is to adjust the air conditioners to maximize users' temperature satisfaction values.

   Under the continuous satisfaction model, we are given the input including $(\mu_i^t, \sigma_i^t, t_i^t)$ for all $i = 1 \ldots n$; d_j^{min} and d_j^{max} for all $j = 1 \ldots m$; $F_{s_k}(d_1, d_2, \ldots, d_m)$ for all $k = 1 \ldots q$. Our goal is to solve d_j, for all $j = 1 \ldots m$ with the objective function:

$$\max \quad \sum_{i=1}^{n} f_i^t \left(\overline{F_{u_i}}(d_1, d_2, \ldots, d_m) \right) \tag{3.19}$$

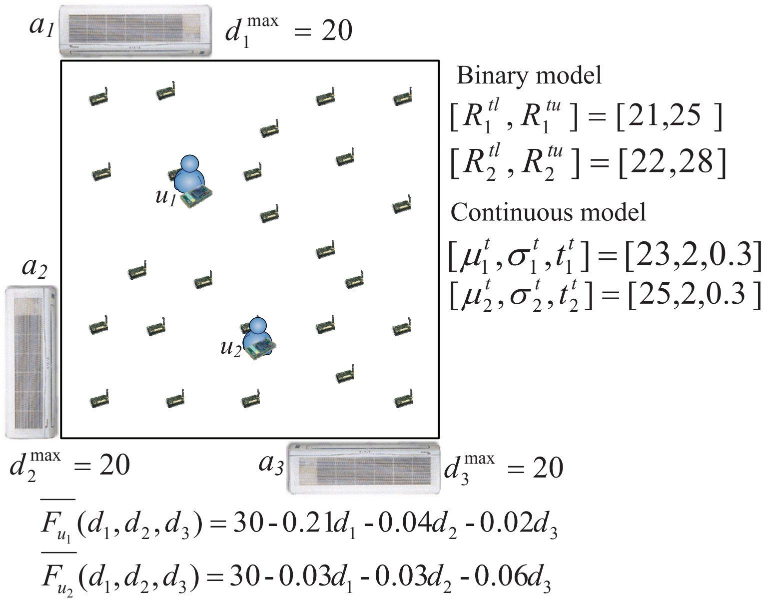


Figure 3.20: The example for binary and continuous satisfaction models.

subject to:

$$\mu_i^t - \sigma_i^t \sqrt{-2ln(t_i^t)} \leq \overline{F_{u_i}}(d_1, d_2, \ldots, d_m) \leq \mu_i^t + \sigma_i^t \sqrt{-2ln(t_i^t)}, \quad \text{for all} \quad i = 1 \ldots n \tag{3.20}$$

$$d_j^{min} \leq d_j \leq d_j^{max}, \quad \text{for all} \quad j = 1 \ldots m \tag{3.21}$$

Eq. (3.19) is to maximize the sum of satisfaction values of all users. Eq. (3.20) imposes that all users' temperature requirements should be met. Eq. (3.21) specifies the bounds. This is a non-linear programming problem and can be solved by a sequential quadratic programming (SQP) method [9]. If there is no feasible solution, we gradually relax users' requirements to make this problem feasible. We propose an iterative process as follows: First, we run the SQP method. If no feasible solution is found, we change u_i's temperature threshold to $\max(0, t_i^t - \beta)$ for each $i = 1 \ldots n$, where β is a constant. Then we run the SQP method again. This is repeated until a solution is found or the retry limits we meet. The results of control levels and corresponding sensor readings should be inserted into database to be an new entry.

Fig. 3.20 shows the example of the binary and satisfaction continuous models. There are three air conditioners a_1, a_2, and a_3 in this environment. For binary satisfaction model, two users u_1 and u_2 have their temperature requirements $[R_1^{tl}, R_1^{tu}] = [21, 25]$ and $[R_2^{tl}, R_2^{tu}] = [22, 28]$, respectively. By several training data, we construct their regression function $\overline{F_{u_1}}(d_1, d_2, d_3) =$

$30 - 0.21d_1 - 0.04d_2 - 0.02d_3$ and $\overline{F_{u_2}}(d_1, d_2, d_3) = 30 - 0.03d_1 - 0.03d_2 - 0.06d_3$ for u_1 and u_2, respectively. The objective function is

$$\min \quad d_1 + d_2 + d_3$$

subject to:

$$\begin{array}{rcccl} 21 & \leq & 30 - 0.21d_1 - 0.04d_2 - 0.02d_3 & \leq & 25 \\ 22 & \leq & 30 - 0.03d_1 - 0.03d_2 - 0.06d_3 & \leq & 28 \\ 0 & \leq & d_1 & \leq & 20 \\ 0 & \leq & d_2 & \leq & 20 \\ 0 & \leq & d_3 & \leq & 20 \end{array}$$

Because this problem is feasible, the solution is $d_1 = 20$, $d_2 = 8.34$, and $d_3 = 18.86$. Similarly, for continuous satisfaction model, two users u_1 and u_2 have their temperature requirements $[\mu_1^t, \sigma_1^t, t_1^t] = [23, 2, 0.3]$ and $[\mu_2^t, \sigma_2^t, t_2^t] = [25, 2, 0.3]$, respectively. Given $t_1^t = 0.3$ and $t_2^t = 0.3$, we can derive $[23 - 2\sqrt{-2\ln(0.3)}, 23 + 2\sqrt{-2\ln(0.3)}] = [19.9, 26.1]$ and $[25 - 2\sqrt{-2\ln(0.3)}, 25 - 2\sqrt{-2\ln(0.3)}] = [21.9, 28.1]$ for u_1 and u_2, respectively. Other configures are the same as binary satisfaction model. The objective function is

$$\max \quad f_1^t(30 - 0.21d_1 - 0.04d_2 - 0.02d_3) + f_2^t(30 - 0.03d_1 - 0.03d_2 - 0.06d_3)$$

$$\begin{array}{rcccl} 19.9 & \leq & 30 - 0.21d_1 - 0.04d_2 - 0.02d_3 & \leq & 26.1 \\ 21.9 & \leq & 30 - 0.03d_1 - 0.03d_2 - 0.06d_3 & \leq & 28.1 \\ 0 & \leq & d_1 & \leq & 20 \\ 0 & \leq & d_2 & \leq & 20 \\ 0 & \leq & d_3 & \leq & 20 \end{array}$$

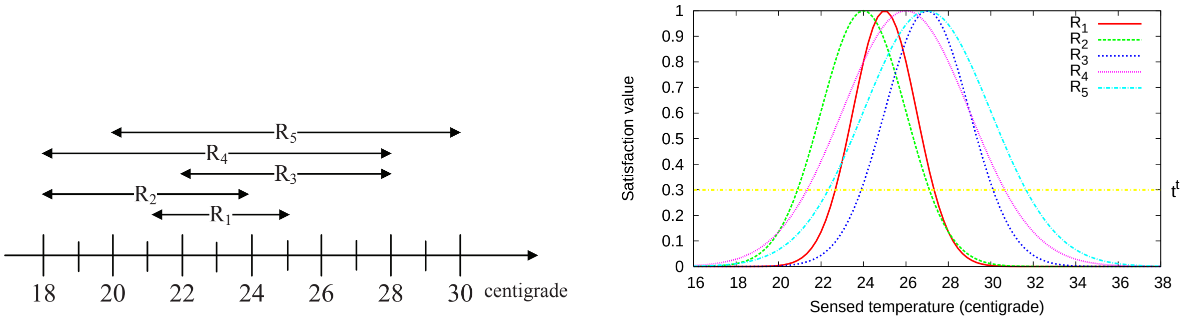
Again, the problem is also feasible. The solution is $d_1 = 8.4$, $d_2 = 19.27$, $d_3 = 19.77$.

3.4.3.3 Extension

Obviously, when executing our system once, it will increase one entry into database. Hence, the data size will be very huge after running a long period of time. Here, we can adopt k-means clustering [40] algorithm to reduce the data size. By k-means clustering, we can get k sets of training data. Then, from each set, we can retain one or more representative entries and discard the others to reduce the size of database.

3.4.4 Evaluations

To understand how our schemes for air conditioners meet users' requirements while saving energy, we have developed a simulation program. Two scenarios are considered. One scenario,



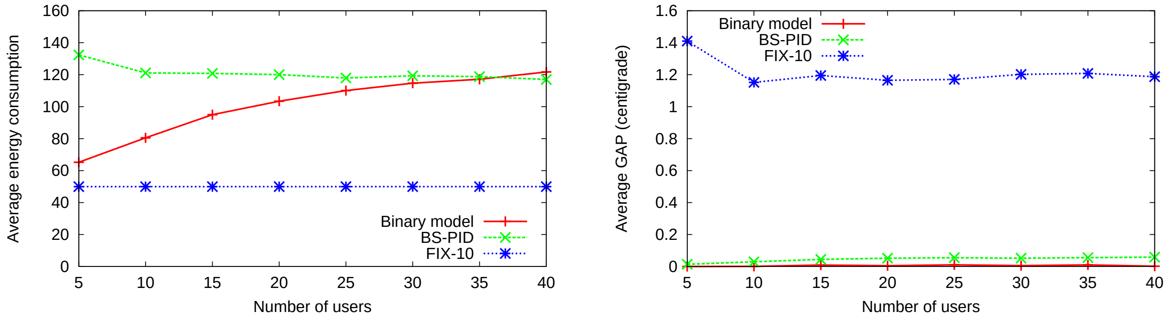
(a) (b)

Figure 3.21: The temperature requirement pools: (a) binary satisfaction model, (b) continuous satisfaction model.

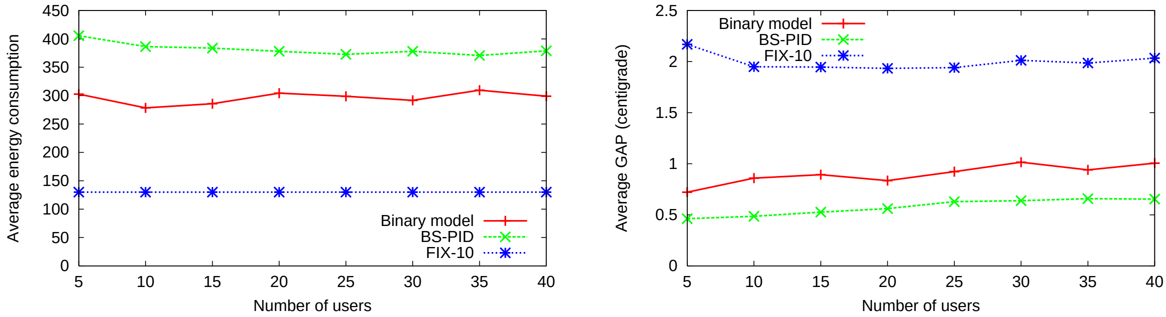
denoted as $S1$, is in a room of size $10 \times 10\ m^2$ with 5 air conditioners. The other scenario, denote ad $S2$, is in a room of size $20 \times 20\ m^2$ with 13 air conditioners. We set $d_j^{min} = 0$ and $d_j^{max} = 40$ for all j. Our proposed algorithms are compared to other two schemes, called FIX and BS-PID. The FIX scheme is a very intuitive one. All air conditioners in the room are set to a fixed level n. We denote this scheme as FIX-n below. The BS-PID scheme is the binary search PID scheme. Each air conditioner selects the nearest one user to satisfy by binary search to find the suitable control level. Once all air conditioner get the control levels, then all air conditioners turn on to that control levels. After a period of time, all sensors report current readings. According to the current readings, all air conditioners redo above search process again. Above steps will be repeated until all users' requirements are satisfied or retry limit we meet. In this simulation, we also record the execution round of above schemes. One round means the temperature achieve the stable state after adjusting the levels of air conditioners. As Section 3.4.1 mentioned, the PID method needs a long time to get the suitable state. Below, we compare the outcomes according to the two satisfaction models introduced in Section 3.2.

- Binary satisfaction model: In this simulation, we define the binary requirement pool for users. As shown in Fig. 3.21(a), each range R_i represents a temperature interval. Each user will randomly choose one interval from the pool as its requirement. We consider two performance indices here. The first index is the total energy consumption. The second index is called GAP, which reflects the difference between the provided temperature and the required one. The GAP for user u_i is

$$GAP(u_i) = \begin{cases} 0 & \text{, if } R_i^{tl} \leq v_{u_i} \leq R_i^{tu} \\ \min(|R_i^{tl} - v_{u_i}|, |R_i^{tu} - v_{u_i}|) & \text{, otherwise.} \end{cases} \quad (3.22)$$



(a)



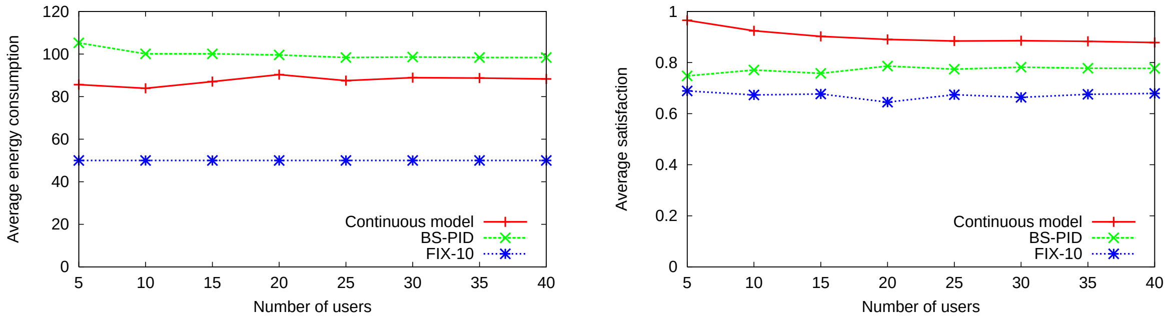
(b)

Figure 3.22: Comparison under the binary satisfaction model: (a) network scenario $S1$, (b) network scenario $S2$.

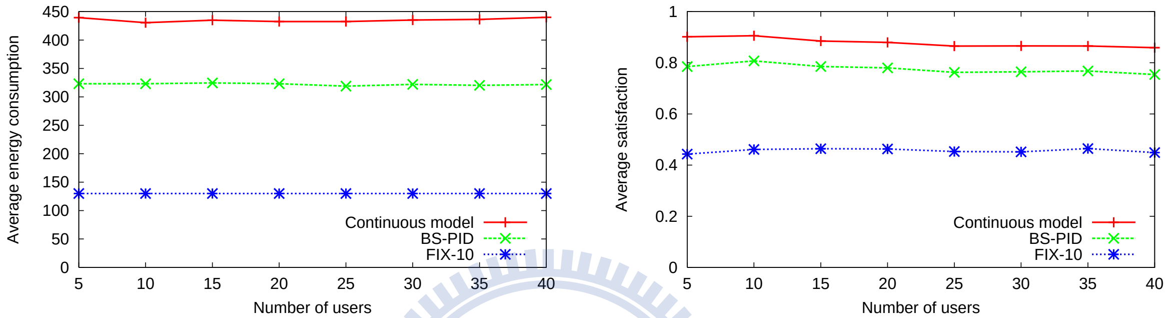
We will measure the average GAP of all users.

Fig. 3.22(a) shows the results in network scenario $S1$. In Fig. 3.22(a), our scheme is energy-efficient while keeping the average GAP smaller than 1 °C. Although the average GAP of BS-PID is very close to our binary model, the execution round is higher than our binary model in Fig. 3.24(a). Hence, the BS-PID needs a long period of time to adjust the air conditioners. Fig. 3.22(b) shows the results in network scenario $S2$. The result is similar to network scenario $S1$. This demonstrates that our scheme is quite scalable to network size.

- Continuous satisfaction model: Similarly, we define the continuous requirement pool for users. As shown in Fig. 3.21(b), each curve represents the continuous temperature requirement. The satisfaction threshold t is set to 0.3. We compare two performance indices: average user satisfaction and energy consumption. Fig. 3.23 shows the results in network scenario $S1$ and $S2$. These results consistently indicate that our scheme provides the highest satisfaction levels and outperforms FIX and BS-PID schemes in energy consumption.

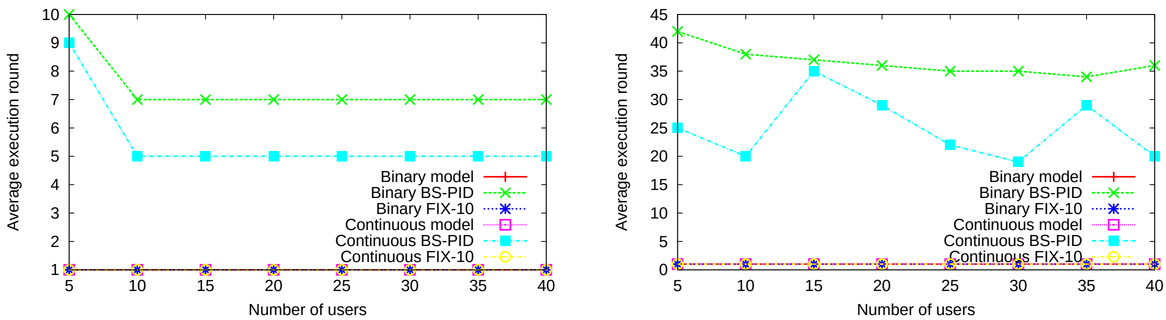


(a)



(b)

Figure 3.23: Comparison under the continuous satisfaction model: (a) network scenario $S1$, (b) network scenario $S2$.



(a) (b)

Figure 3.24: Comparison the execution round under binary and satisfaction models: (a) network scenario $S1$, (b) network scenario $S2$.

Chapter 4

A Two-tier Location Platform for a Smart Living Space

4.1 System Overview

Indoor mobile robots have lots of potential applications in daily life, such as home care, room cleaning, object finding, and emergency supporting [8][58][59]. Most mobile services would require a robot to determine its current location [4][12][28][36][38][39]. While this issue has been intensively studied in factory environments [4][54], 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 chapter is to design a suitable indoor navigation mechanism for mobile robots to meet this need.

To precisely navigate a robot, a localization method is needed [7][31][44]. GPS dominates many outdoor localization applications. However, for indoor localization, a globally usable solution is still missing. In Section 4.2, we will review some existing indoor localization solutions for mobile robots. In this chapter, we propose a framework to design a "self-guided" indoor robot. Our goal is to take advantage of existing technologies, such as RFID, wireless networks, and indoor localization systems to achieve this goal. We list some design guidelines of our framework:

- Plug-and-play: It is desirable that the robot can automatically start without going through any setup effort. For example, some robotic systems [16][17] need a training phase before getting started, but some [12][28][36] need not. The robots in [12][28][36] do not require to have any pre-knowledge about the environment before entering the environment.

- Self-guided: The robot should be able to guide itself to any location that it desires to and can still recalibrate itself. For example, the robots in [4][38][39] need not to know their initial locations when entering an environment.

- Multi-level localization accuracy: Depending of different situations, different levels of accuracy, perhaps at different costs, can be provided. An example is [47].
- Low or no extra infrastructure: The extra infrastructures required to guided the robot should be as minimal as possible. Also, we may use existing infrastructures to achieve this goal.
- Low cost: we are not looking for complicated systems. The system should be inexpensive and affordable.
- Scalability: Our system should be scalable to larger environments without adding much overheads.

Following these guidelines, we propose a two-tier localization architecture for the design of a self-guided robot. Our system does not rely on building a new infrastructure. Instead, we utilize existing infrastructures, such as RFID and WiFi networks, which might be widely and densely deployed in future homes/offices. Assuming that the robot is equipped with a WiFi interface and a RFID reader, our system supports localization at two precision levels. A location server based on a WiFi fingerprinting method [7][31] helps the robot to position itself at meter-level precision. Also, on the ground, some RFID tags are deployed and a tag map is pre-installed in the location server to help the robot to position itself at centimeter-level precision. Through this two-tier mechanism, we can meet the above requirements at reasonable low cost and deployment overhead. The result should be applicable to many indoor navigation services in home/office environments. We also design a spiral search algorithm to help the robot to identify a RFID tag and analyze the corresponding cost (spiral moving distance) that a RFID tag is expected to be found.

4.2 Preliminaries

4.2.1 RFID and WiFi

RFID (Radio Frequency IDentification) generally refers to systems that can transmit identity information of objects via wireless technologies. Such technologies have been widely applied to stock tracking, security, logistics, etc. A typical RFID system is composed of tags, readers, and servers. A tag is a tiny radio device with a simple silicon microchip attached to a small flat antenna mounted on a substrate. RFID tags have two types: *active* and *passive*. An active tag has its own power source and thus has a longer transmission distance. A passive tag, on the contrary, has no power source and thus can only respond to a reader when accumulating sufficient energy in its capacitor. Readers can send and receive RF signal to and from tags. Typical operation frequencies include 455 MHz, 2.45 GHz, and 5.8 GHz. Servers can be general computers to interact with readers to provide various applications.

The term WiFi stands for Wireless Fidelity. WiFi is a class of wireless local area network (WLANs) based on IEEE 802.11 standards [22]. Nowadays, it is widely used for broadband Internet access. A WiFi network consists of some access points (APs) and WiFi interfaces at clients. Recently, WiFi APs have been intensively studied for providing indoor localization when GPS signal is not available. In this chapter, we will apply such techniques for both guiding our robot and providing "RFID tag maps" on request of the robot.

4.2.2 Localization Techniques

Localization solutions can be classified as AoA-based [43], ToA-based [1], TDoA-based [50], infrared-based [11], ultrasonic-based [6], RF-based [7], and fingerprint-based [7][31][44]. In this work, observing that WiFi networks are widely deployed, we are more interested in applying fingerprinting localization methods [7][31], which rely on existing WiFi APs as beacons. Such solutions have two phases. In the *training phase*, radio signal strength (RSS) patterns are collected at a set of training locations from pre-deployed beacons and stored in a database (called radio map). During the *positioning phase*, a device 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. Several variants of fingerprinting techniques have been proposed [31].

4.2.3 Robot Navigation Techniques

For most mobile robot applications, it is essential to precisely locate and navigate a robot. Robot navigation techniques can be categorized as laser-based, track-based, and vision-based. In [38], three or more laser guiding beams are used to determine a robot's location. A track-based system uses some traces, such as magnetic tapes on the ground, to guide a robot [4]. Vision-based systems [39] are more flexible, but they require high computing resources and are constrained by line-of-sight.

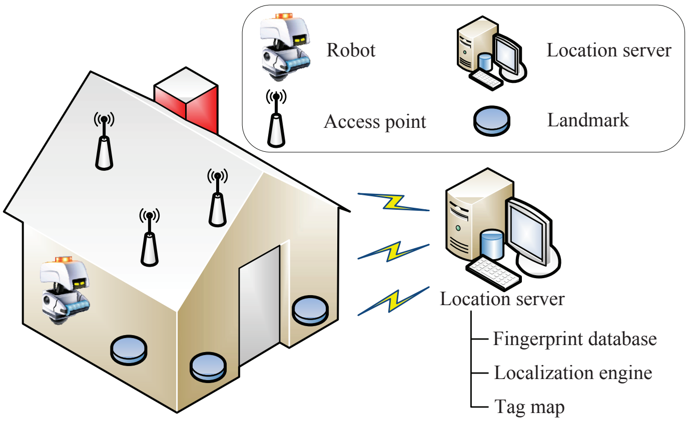


Figure 4.1: The system architecture of our self-guided robot system.

Recently, RFID techniques have also been applied to robot navigation [12][28][36]. In [12][36], an array of passive RFID tags are deployed on the entire field. Each tag represents a unique location. On scanning any tag, the robot can calculate its location. Grid deployment is discussed in [12], while triangular deployment is discussed in [36]. Such systems normally require a large number of RFID tags. The work [28] uses a "rotatable" RFID reader to guide a robot to a stationary target with an active tag. An AoA-like model is adopted. This solution is more costly and is mainly for one single target. Our system uses sparse passive tags for centimeter-level positioning and it takes advantage of existing WiFi networks for meter-level positioning.

4.3 Design of a Two-Tier Self-Guided Robot

4.3.1 System Architecture

Fig. 4.1 shows the system architecture of our self-guided robot system. There are four main components:

- WiFi networks: The networks contain a number of access points, which periodically transmit beacons. Through these beacons (as reviewed in Section 4.2), a device can estimate its location at a meter-level accuracy through a fingerprinting mechanism.

- Location server: The location server runs a fingerprinting localization algorithm, such as those in [7][31]. It has three modules: (i) fingerprint database, (ii) localization engine, and (iii) tag map. The fingerprint database contains a set of training data, each binding a physical location in the target field with a vector of RSS patterns that can be received

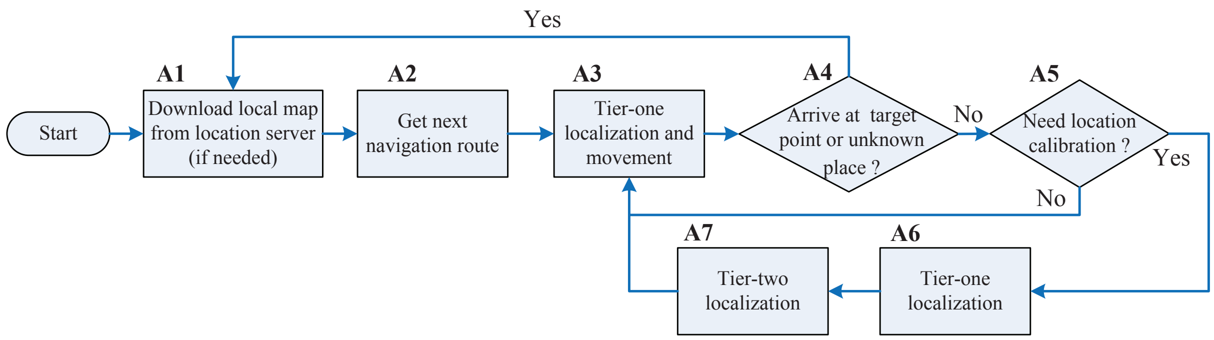


Figure 4.2: Flow chart of our navigation procedure.

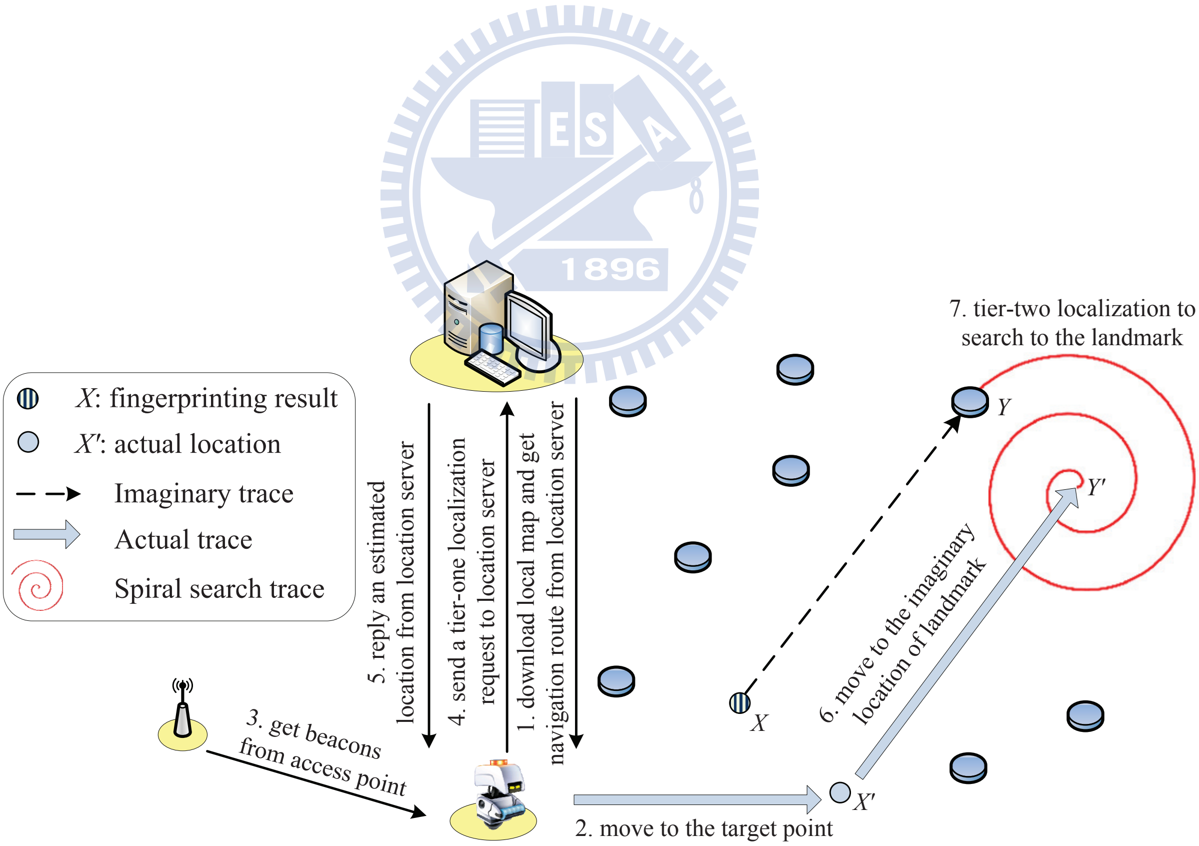


Figure 4.3: An example of our two-tier localization.

at that location. When localization is needed, the robot can collect its current RSS pattern and inquire the localization engine to determine its location by comparing to the fingerprint database (which we call meter-level positioning). On getting its location, the robot can further inquire the local tag map, which contains the locations of the landmarks deployed nearby the robot.

- Landmarks: These landmarks are passive, each of which has been registered in the tag map to identify a unique location. Finding any landmark is also considered a localization event (which we call centimeter-level positioning). These landmarks are divided into groups and each group of landmarks are deployed in a special shape (to be elaborated later on) to facilitate the robot to discover them. These tag groups may be deployed in a regular manner and at main turning points or service points.

- Robot: The robot is a mobile platform equipped with an e-compass, a WiFi interface, and a RFID reader. It conducts our two-tier localization scheme to determine its current location. In the first tier, it only needs to inquire the location server. In the second tier, it needs to search the tag map for landmarks and we will propose our searching strategy later on. The e-compass helps indicate the orientation of the robot. As to service provisioning, it may be equipped with other tools, such as cameras, arms, etc. (this is out of the scope of this work).

4.3.2 Two-Tier Localization Scheme

Fig. 4.2 shows the flow chart of our two-tier localization scheme for the robot. One fact for most motor systems is that a robot is subject to a certain degree of errors when it moves a certain amount of time or distance. Furthermore, moving on different types of floor materials (which is quite common in home/office environments) may also incur different degrees of errors. Below, we describe the detail procedures, which take the above errors into account.

A1. To initialize, the robot will download the local map from the location server (this can be done through the WiFi interface). Since we target at indoor applications, the map should include the floor plan and the interior design. Whenever the robot finds itself entering a new environment (e.g., by finding new WiFi APs), it can download a new map.

A2. After getting the local map, the robot will get the next navigation route and command from the location server.

A3. The robot then tries to move to the next target point. Depending on the tolerable error of the robot's motor system, the robot will enter A4 as necessary (e.g., it may need to calibrate its location after moving a certain distance or a certain amount of time). Also, it may enter A4 if it suddenly finds itself in an unfamiliar environment (e.g., it may find some unfamiliar WiFi APs).

A4. The robot checks if it has arrived at the target point or an unfamiliar environment. If so, it goes to A1; otherwise, it goes to A5. (Note that the check is application-dependent. It may be triggered, for example, by landing at a special landmark, contacting a charger, receiving a special signal, etc.).

A5. The robot then checks if it needs to calibrate its current location. If so, it goes to A6; otherwise, it goes to A3. (Note that the check may be done by conducting a tier-one localization. If the result deviates from its expected location by a certain threshold or above, a calibration is needed.)

A6. In this tier-one localization, the robot collects its current RSS patterns and inquires its location with the location server. Based on the result, it also asks for a local tag map from the location server.

A7. In this tier-two localization, the robot tries to scan nearby landmarks. Here, we propose to use a spiral search scheme (see Section 4.3.3). On tracking any landmark, the robot knows its accurate location. Then, it calibrates its orientation by its e-compass. Then, go to A3.

Fig. 4.3 shows an example of the above procedure. In step 1, the robot will download the local map and tag map and get the navigation route from the location server. In step 2, it tries to move to the target point. When the robot thinks that it has arrived at the expected target point, it checks if it needs to recalibrate itself. In step 3, it collects gets beacons from nearby access points and sends a tier-one localization request to the location server. In step 4 and 5, it gets an estimated location X at a meter-level precision from the location server. Since the fingerprinting localization is subject to error, let us suppose that its actual location is X'. In step 6, the robot starts to move to the landmark Y. Its actual trace is shown by the bold arrow, while it thinks that its trace is the dotted arrow. At point Y', the robot is unable to find the landmark

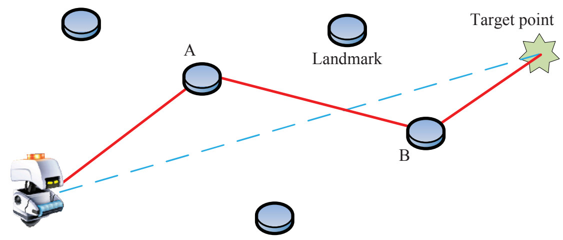


Figure 4.4: An example of choosing intermediate landmarks.

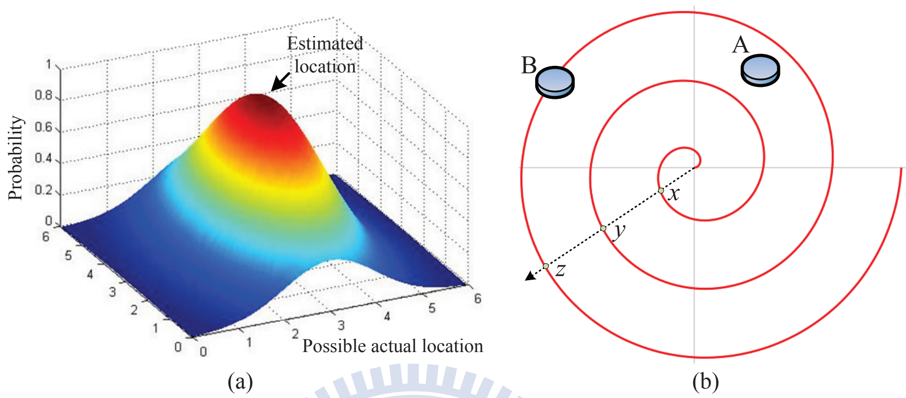


Figure 4.5: (a) Typical probability distribution of the actual location obtained by the fingerprinting localization. (b) An example of spiral search.

Y. Hence, it starts a tier-two localization using a spiral search to find the landmark Y in step 7. Once the landmark is found, the robot positions itself at centimeter-level precision.

Also note that in A3, the robot may not need to move in a straight line in order to get to the next target point. To avoid accumulating too much error while moving, it may choose to land on some intermediate landmarks before reaching the target point. The problem of inserting appropriate intermediate landmarks can be done by a geographic forwarding algorithm, such as those in [27][65], where one can continuously choose the next landmark which is within a tolerable distance from its current location and is nearest to the target point. Fig. 4.4 shows an example, where landmarks A and B are selected as intermediate points.

4.3.3 Spiral Search Scheme

In our tier-two localization, a search mechanism is needed to find a target landmark in step A7. We propose to use a spiral search because it can gradually expand its search range. The problem is modeled as follows. After the fingerprinting localization, the PDF of the actual location of the robot with respect to the estimated location is typically modeled by a bell-shape 2D Gaussian

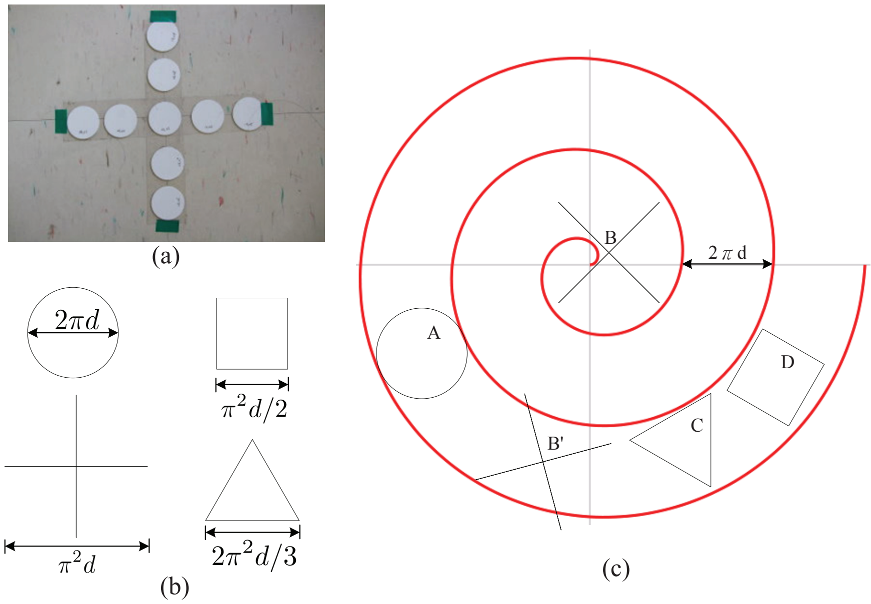


Figure 4.6: Different shapes of RFID tag groups and their intersection properties.

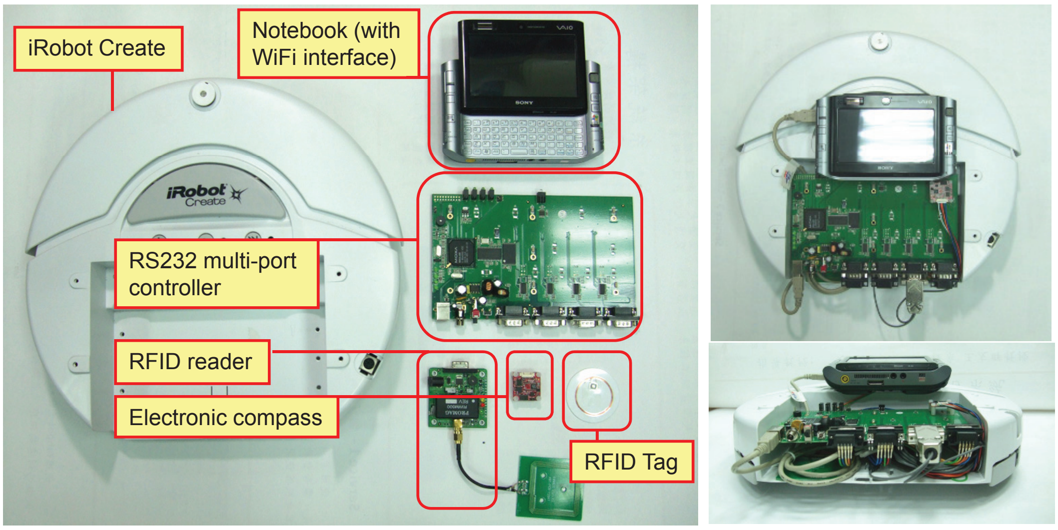
distribution, as illustrated in Fig. 4.5(a). Hence, we propose to adopt the Archimedean spiral [3], which has successive turns at a constant separation, as illustrated in Fig. 4.5(b), to search the robot's surrounding.

Without loss of generality, let the current location of the robot at the beginning of the search be the origin. The Archimedean spiral can be modeled by the polar coordinate

$$r = d\theta ,$$

where d is a constant. It is not hard to see that for any vector leaving from the origin, any two consecutive intersecting points of the vector and the spiral have a distance of $2\pi d$ (for example, the distance between points x and y in Fig. 4.5(b) is $2\pi d$, and the same for that between y and z).

After leaving the spiral origin, the robot will try to scan any RFID on the ground (whenever a tag is found, the tier-two localization succeeds). In the example of Fig. 4.5(b), the robot will miss the RFID tag A, but will find the tag B (note that successfully finding a tag is constrained by the sensing distance of the RFID reader). Because tags are quite cheap, to facilitate the search process, we propose to deploy RFID tags in groups, each of a certain shape. In Fig. 4.6(a), we show a square-shape tag group. In Fig. 4.6(b), we show several other shapes of tag groups (circle, cross, and triangle). Along the circumference of a shape, we place RFID tags uniformly, each separated by a fixed distance. Clearly, different shapes will require different numbers of



(a) (b)

Figure 4.7: (a) Hardware components of our self-guided robot. (b) The self-guided robot after integration.

RFID tags (and thus different costs).

No matter which shape is used, we need to guarantee that for any orientation of a tag group, it will intersect with the Archimedean spiral. Let $r = d\theta$ be the spiral. Recall that the spiral line is separated by a distance of $2\pi d$ in any location. Let us use a circle tag group with a diameter $2\pi d$ as a basis. Since its perimeter is $2\pi^2 d$, using the same perimeter, Fig. 4.6(b) shows the side lengths of different tag group shapes ($\pi^2 d$, $2\pi^2 d/3$, and $\pi^2 d/2$ for cross, equilateral triangle, and square, respectively). Unfortunately, as shown in Fig. 4.6(c), only the circle shape guarantees the intersection property; the other shapes B, C, and D all fail to guarantee this property at certain orientations. In particular, we note that the cross shape B only fails to intersect with the spiral nearby the origin of the spiral; after leaving the origin, the intersection property can be guaranteed at all orientations (see B' in the figure).

It would be interesting to calculate the expected time that the robot can locate itself after starting a search. Assuming that the robot moves at a constant velocity and the error of the tier-one localization follows a 2D Gaussian distribution, we derive the following results.

Lemma 1. *Given an Archimedean spiral $r = d\theta$ and a circle of radius E centered at the origin, they intersect at only one point and the length of the spiral from the origin to intersection point is*

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

Proof. Given any θ value, the arc length of the Archimedean spiral from the original to point (r, θ) is

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

Since $E = r = d\theta$, replacing θ by $\frac{E}{d}$, we have proved this lemma. □

From Lemma 1, the expected value of our tier-two localization can be decided.

Theorem 1. *Assuming that the error of the tier-one localization follows a* $(0, \sigma^2)$ *Gaussian distribution such that the probability of the error distance* E *is* $f(E) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-E^2}{2\sigma^2}}$*, the expected search distance of tier-two localization is*

$$\int_0^\infty len(\frac{E}{d})f(E)dE.$$

4.4 Prototype and Performance Evaluation

At the Eco-City Lab [15] in the National Chiao Tung University, we have developed a prototype of proposed self-guided robot. In this section, we report our prototyping experiences and some performance evaluation results.

4.4.1 Hardware/Software Components and Design Issues

Fig. 4.7 shows the components of the self-guided robot. We adopt iRobot Create [25] as the mobile platform and attach a RFID reader, an electronic compass, and a notebook with a WiFi interface to the iRobot. The iRobot Create is a programmable moving platform with flexibility for extension. It has a cargo bay with a 25-pin port that can be used for digital and analog input and output. We connect an RS232 multi-port controller on the cargo bay. The RFID reader operates under frequency 13.56 MHz and has an acceptable reading range of 5 cm. The iRobot, RFID reader, and electronic compass are connected by the RS232 multi-port controller, through which, the robot can get its (tier-two) location and orientation information. We adopt the cross-shape (30 cm $\times$ 30 cm) RFID tag group as shown in Fig. 4.6(a) in our implementation. Each tag contains a unique ID and its coordinate is pre-registered to the system.

Fig. 4.8 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

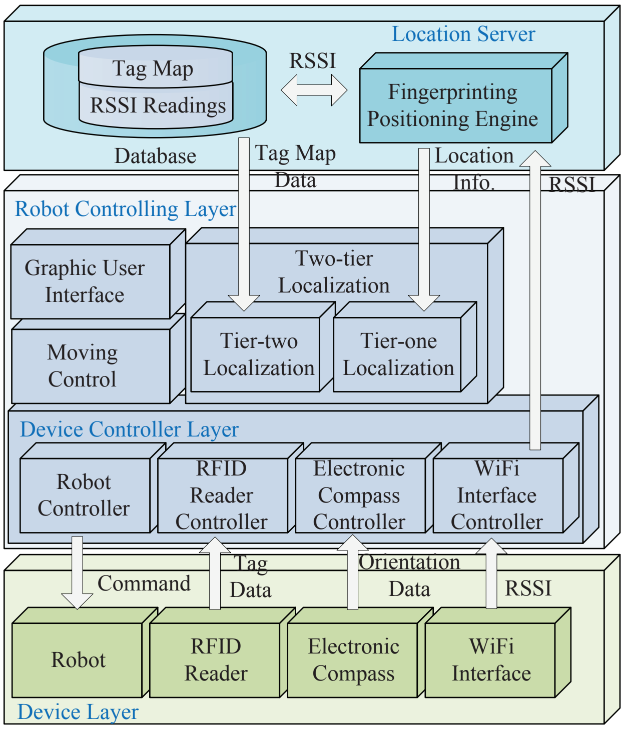


Figure 4.8: The functional blocks of implementation.

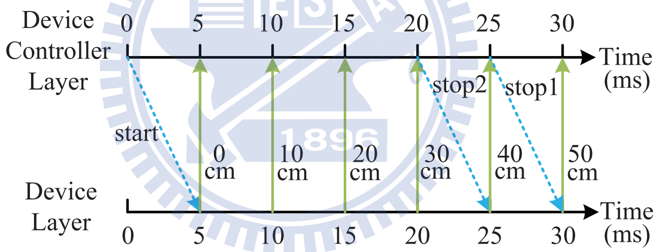


Figure 4.9: An example of communication delay.

Robot Controlling Layer contains the Two-tier Localization scheme and Moving Control mechanism which can get location information and tag map data from Location Server. The Location Server contains a database which stores a set of RSSI readings and RFID tag map for tier-one and tier-two localization, respectively.

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

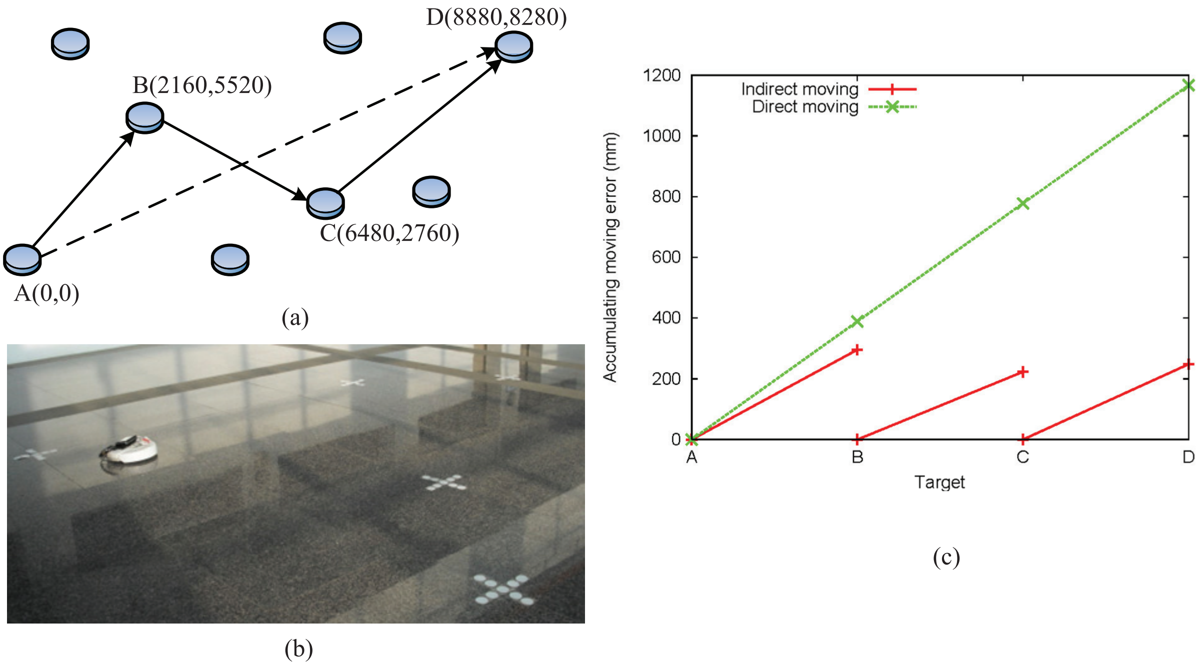


Figure 4.10: The long distance moving experiment of self-guided robot.

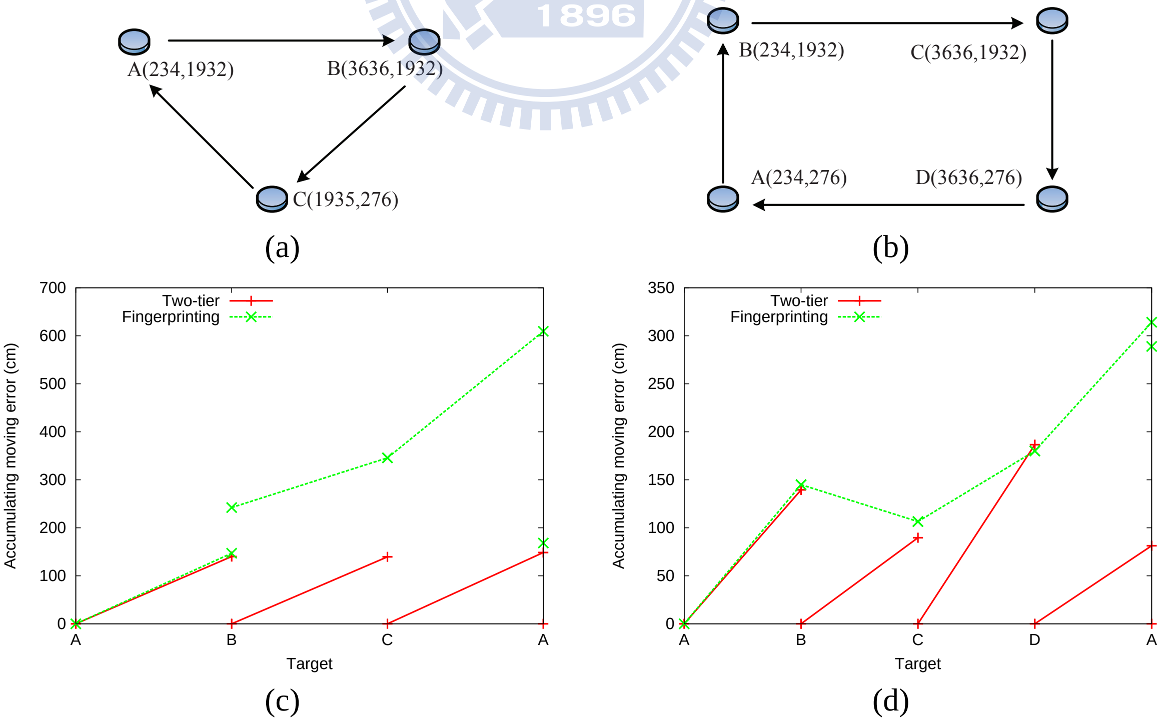


Figure 4.11: The experiment of two-tier localization scheme.

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 recording values beforehand to relieve the influence of communication delays. Fig. 4.9 shows an example. Here, we send a start command to trigger robot moving. The 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 meters accumulating 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 proportion 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 does not get any information. We only record the tag type, landmark group ID, offset value, X and Y axis in the RFID tags.

4.4.2 Evaluations

In this section, we evaluate our self-guided robot by two experiments. The first one is to evaluate the long distance moving. The second one is to evaluate the two-tier localization scheme.

Section 4.3.2 mentions that we will insert some intermediate points while the robot moves long distance. As shown in Fig. 4.10(a) and Fig. 4.10(b), we place several RFID tags on the ground and the number in parentheses are the location of the RFID tag. We compare the robot moves from point A to point D, denoted as direct moving, and from A to B, C, and D, denoted as indirect moving. Fig. 4.10(c) shows that indirect moving can avoid larger accumulating error than direct moving. Hence, if we apply the self-guided robot to the high accuracy application, we should insert more intermediate points to reduce the accumulating moving error.

Next, we evaluate the two-tier localization scheme against the WiFi fingerprinting localization algorithm [7]. As shown in Fig. 4.11(a) and Fig. 4.11(b), we deploy the RFID tags as triangle and rectangle, respectively. The robot traverses these RFID tags in order by two-tier and fingerprinting localization schemes. For fingerprinting algorithm, we define a threshold value $\delta = 390$ cm. When the robot moves to the target, it will relocate itself by fingerprinting algorithm if the accumulating moving error greater than δ. Fig. 4.11(c) and Fig. 4.11(d) show the result of triangle and rectangle deployments, respectively. The two-tier localization scheme can guarantee the robot guided itself and the moving error is lower than fingerprinting scheme.

4.4.3 Case Studies

Our system proposes a robot can guide itself in the 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.

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 can apply our system in this scenario. We deploy landmark RFID tags, exhibit RFID tags, access points, location server and self-guided robot in the exhibition. The landmark RFID tags and access points are used to tier-two and tier-one localization, respectively. Except the landmark RFID tags, 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-guided robot can guide the users to visit the exhibition and also can dynamically plan the visiting routes according to users' preference exhibits.

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 [7][31][44]. 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

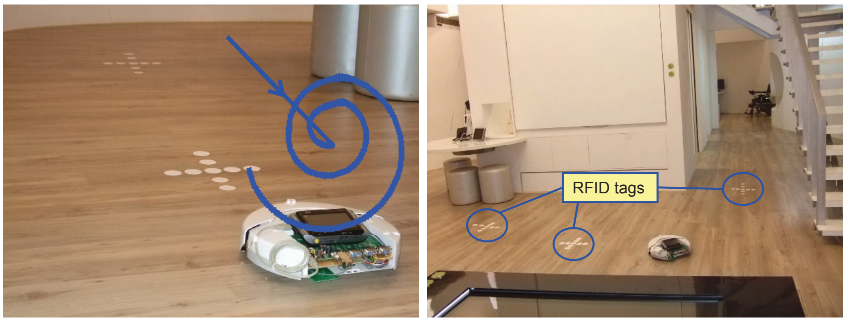


Figure 4.12: An implementation scenario of our self-guided robot.

[67]. Manually calibrating radio maps is error-prone. Besides, environment may change, furniture may be moved, and beacons may be reconfigured or upgraded anytime [30]. The radio signal strength may vary at different times. This may result in non-negligible errors when positioning objects. Hence, we adopt the self-guided 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. Fig. 4.12 shows our implementation scenario in Nation Chiao Tung University Eco-City Lab [15].

Chapter 5

Conclusions and Future Directions

This dissertation contain four works. The first one is an energy conservation system which construct a framework of smart living space. The second one is the light control system considering users' lighting requirements and energy consumption. The third one is the air conditioners control system considering users' temperature requirements and energy consumption. The last one is the mobile services platform to provide indoor location-based services.

In Chapter 2, we propose an iPower system designed for energy conservation in an intelligent building and provision of personalized services for environment control. The iPower system can detect if there is possible waste of electricity by WSNs and then turn off these unnecessary electric appliances via the power-line control devices with a user-friendly design. The iPower system also provides personalized services in which electric appliances can be automatically adjusted to satisfy users' requirements. We have presented the design and implementation details of iPower. Prototyping experiences and design issues are also given in this work.

In Chapter 3, we propose two control systems in Section 3.3 and Section 3.4, which consider both users' requirements and energy consumption. In Section 3.3, we present an autonomous light control system. Both whole and local lighting devices are considered. For controlling whole lighting devices, two decision algorithms are proposed. For controlling local lighting devices, a novel surface-tracking scheme is proposed. Our system can dynamically adapt to environment changes and improve our earlier works by eliminating the requirement of tracking users' current locations. We evaluate our algorithms by simulations under different configurations. Besides, we have developed prototypes and deployed the whole system in a room to verify its effectiveness under real conditions. The results do show high consistency. In Section 3.4, we present an intelligent air conditioners control system. Through the training data, we construct the temperature models by regression functions. Then, similar to light control, two models are considered and two control mechanisms are proposed to adjust the air conditioners quickly and satisfy users' temperature requirements. The simulation results show that our schemes can quickly adjust the air conditioners to satisfy users' requirement and achieve energy saving.

In Chapter 4, we propose a two-tier localization architecture for indoor mobile robots. The indoor mobile robot can guide itself via our proposed two-tier localization scheme. We also discuss the different shapes of RFID tag groups and the expected searching distance of the robot. Besides the theoretical results, we also implement the whole system and evaluate our system via several experiments. Finally, we propose two case studies to apply our system in our daily life.

Based on the results presented above, several issues which can be discussed and improved are summarized as follows.

- Due to environmental noises or errors, the readings of sensor nodes may not be accurate. This may miss lead the control server to make incorrect decisions. To solve this problem, we may apply the solutions in [10][53][68] to alleviate the effects of these inaccurate sensing readings or improve above works to get more accurate sensing values.

- For the iPower, light control, and air conditioners control systems, there is a basic limitation in our system. We enforce that users must carry sensors to measure their current light or temperature values. This is due to the nature of light or temperature propagation. Without these sensors, our system can not get any information nearby the users. There are some alternatives, such as using motion sensors or thermal sensors to detect users' locations or rough light environments, respectively. Unfortunately, such approaches still can not provide satisfactory solutions so far.

- For light and air conditioners control systems, we adopt Gaussian functions to represent users' satisfaction levels. However, the utility to human, in terms of light intensity or temperature value, is still an unknown factor. This may deserve further study in the medical science field.

- For light control and air conditioners control systems, we only use the light intensities of lighting devices and control levels of air conditioners to represent the energy consumption. In the future, we will develop the wireless power meters to measure the energy consumption in practice. Through the wireless power meters, our control server can monitor the actual power consumption of lighting devices and air conditioners.

Bibliography

[1] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, 2001.

[2] R. Alcala, J. Benitez, J. Casillas, O. Cordon, and R. Perez. Fuzzy control of hvac systems optimized by genetic algorithms. *Applied Intelligence*, 18:155–177, 2003.

[3] Archimedean spiral, http://en.wikipedia.org/wiki/Archimedean_spiral.

[4] Automated guided vehicle, http://en.wikipedia.org/wiki/Automated_Guided_Vehicle.

[5] AX-12, Dynamixel series robot actuator, http://www.crustcrawler.com/motors/AX12/index.php.

[6] 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.

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

[8] R. Barea, L. Bergasa, E. Lopez, M. Ocana, D. Schleicher, and A. Leon. Robotic assistants for health care. In *Proc. of IEEE Int'l Conference on Robotics and Biomimetics*, 2008.

[9] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 45(1):1–51, 1995.

[10] J. Branch, B. Szymanski, C. Giannella, R. Wolff, and H. Kargupta. In-network outlier detection in wireless sensor networks. In *Proc. of Int'l Conference on Distributed Computing Systems (ICDCS)*, 2006.

[11] E. Brassart, C. Pegard, and M. Mouaddib. Localization using infrared beacons. *Robotica*, 18(2):153–161, 2000.

[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] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to algorithms, 2001.

[14] S. K. Das, D. J. Cook, A. Battacharya, E. O. H. III, and T. Y. Lin. The role of prediction algorithms in the mavhome smart home architecture. *IEEE Wireless Communications*, 9(6):77–84, 2002.

[15] Eco-City, http://www.ecocity.org.tw/.

[16] V. Ganapathy, S. C. Yun, and H. Joe. Neural q-learning controller for mobile robot. In *Proc. of IEEE/ASME Int'l Conference on Advanced Intelligent Mechatronics*, 2009.

[17] V. Ganapathy, S. C. Yun, and J. Ng. Fuzzy and neural controllers for acute obstacle avoidance in mobile robot navigation. In *Proc. of IEEE/ASME Int'l Conference on Advanced Intelligent Mechatronics*, 2009.

[18] O. Gassmann, H. Meixner, J. Hesse, and W. Gopel. Sensors in intelligent buildings: Sensors applications. *Wiley-VCH*, 2001.

[19] H.-W. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *ACM/Kluwer Mobile Networks and Applications*, 7(5):341–351, 2002.

[20] M. HAENGGI. Mobile sensor-actuator networks: opportunities and challenges. In *Proc. of IEEE Int'l Workshop on Cellular Neural Networks and Their Applications*, 2002.

[21] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: a programmable pervasive space. *IEEE Computer*, 38(3):50–60, 2005.

[22] IEEE standard for information technology-telecommunications and information exchange between systems - local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications, 2007.

[23] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks specific requirements part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs), 2003.

[24] INSTEON, a smart labs technology, http://www.insteon.net/.

[25] iRobot, http://store.irobot.com/corp/index.jsp.

[26] Jennic, JN5139, http://www.jennic.com/.

[27] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proc. of ACM Int'l Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[28] M. Kim and N. Y. Chong. RFID-based mobile robot guidance to a stationary target. In *Mechatronics*, 2007.

[29] Kleinrock. *Queueing systems, volume 1: theory*. John Wiley & Sons, Inc., 1999.

[30] 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*, 8(10):1326–1338, 2009.

[31] 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*, 20(5):678–684, 2008.

[32] LED lamp, http://en.wikipedia.org/wiki/LED_lamp.

[33] J. Liang and R. Du. Thermal comfort control based on neural network for hvac application. In *Proceedings of the 2005 IEEE Conference on Control Applications*, 2005.

[34] Light-emitting diode (LED), http://en.wikipedia.org/wiki/LED.

[35] Light sensor, TSL230, http://www.taosinc.com/.

[36] 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.

[37] C. Lin, C. C. Federspiel, and D. M. Auslander. Multi-sensor single-actuator control of hvac systems. In *Proceedings of the Second International Conference for Enhanced Building Operations*, pages 14–18, 2002.

[38] K. Lingemanna, A. Nuchtera, J. Hertzberga, and H. Surmannb. High-speed laser localization for mobile robots. *Robotics and Autonomous Systems*, 51(4):275–296, 2005.

[39] 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*, 55(10):769–784, 2007.

[40] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[41] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of ACM Int'l Workshop on Wireless sensor networks and applications (WSNA)*, 2002.

[42] Matlab Builder for Java, http://www.mathworks.com/products/javabuilder/.

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

[44] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Chen. Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. *IEEE Trans. on Knowledge and Data Engineering*, 18(9):1181–1193, 2006.

[45] M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, and Y.-C. Tseng. A WSN-based intelligent light control system considering user activities and profiles. *IEEE Sensors Journal*, 8(10):1710–1021, 2008.

[46] H. Park, M. B. Srivastava, and J. Burke. Design and implementation of a wireless sensor network for intelligent light control. In *Proc. of Int'l Symposium on Information Processing in Sensor Networks (IPSN)*, 2007.

[47] J. Y. Park and H. Y. Song. Multilevel localization for mobile sensor network platforms. In *Proc. of Int'l Multiconference on Computer Science and Information Technology*, 2008.

[48] PID controller, http://en.wikipedia.org/wiki/PID_controller.

[49] J. Sankaran. A note on resolving infeasibility in linear programs by constraint relaxation. *Operations Research Letters*, 13(1):19–20, 1993.

[50] 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.

[51] H. Schulzrinne, X. Wu, S. Sidiroglou, and S. Berger. Ubiquitous computing in home networks. *IEEE Communications Magazine*, 41(11):128–135, 2003.

[52] Sebesta. *Concepts of programming languages, 4th ed.* Addison-Wesley, 1999.

[53] B. Sheng, Q. Li, W. Mao, and W. Jin. Outlier detection in sensor networks. In *Proc. of ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2007.

[54] L. Shengfang and H. Xingzhe. Research on the agv based robot system used in substation inspection. In *Proc. of Int'l Conference on Power System Technology*, 2006.

[55] V. Singhvi, A. Krause, C. Guestrin, J. H. Garrett, and H. S. Matthews. Intelligent light control using sensor networks. In *Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, 2005.

[56] SIP: Session initiation protocol, ietf rfc 3261, 2002.

[57] Smarthome, x10-1132b, http://www.smarthome.com.

[58] K.-T. Song, C.-Y. Tsai, F.-S. Huang, J.-W. Hong, C.-Y. Lin, C.-W. Chen, and Z.-S. Lin. Development of the robot of living aid: Rola. In *Proc. of IEEE Int'l Conference on Automation and Logistics*, 2008.

[59] T. Taipalus and K. Kosuge. Development of service robot for fetching objects in home environment. In *Proc. of IEEE Int'l Symposium on Computational Intelligence in Robotics and Automation*, 2005.

[60] TFT LCD, ILI9221, http://www.ilitek.com/products.asp.

[61] Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, and Y.-Y. Hsieh. iMouse: An integrated mobile surveillance and wireless sensor system. *IEEE Computer*, 40(6):60–66, 2007.

[62] X. Wang, J. S. Dong, C. Chin, S. Hettiarachchi, and D. Zhang. Semantic space: An infrastructure for smart spaces. *IEEE Pervasive Computing*, 3(3):32–39, 2004.

[63] Y.-J. Wen, J. Granderson, and A. M. Agogino. Towards embedded wireless-networked intelligent daylighting systems for commercial buildings. In *Proc. of IEEE Int'l Conference on Sensor Networks, Ubiquitous, and Trust-worthy Computing*, 2006.

[64] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Proc. of European Workshop on Sensor Networks (EWSN)*, 2005.

[65] 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.

[66] Worldwideweb consortium (w3c), extensible markup language (xml) 1.0, http://www.w3.org/tr/rec-xml/.

[67] J. Yin, Q. Yang, and L. M. Ni. Learning adaptive temporal radio maps for signal-strength-based location estimation. *IEEE Trans. on Mobile Computing*, 7(7):869–883, 2008.

[68] Y. Zhuang, L. Chen, X. S. Wang, and X. Lian. A weighted moving average-based approach for cleaning sensor data. In *Proc. of Int'l Conference on Distributed Computing Systems (ICDCS)*, 2007.

Curriculum Vitae

Lun-Wu Yeh was born in Hualien, Taiwan, in 1981. He received his B.S. and M.S. degrees in Department of Computer Information Science from the National Chiao Tung University, Taiwan, 2003 and 2005, respectively. He obtained his Ph.D. in Computer Science from the National Chiao Tung University, Taiwan, in October of 2010. His research interests include smart living space and wireless sensor networks.
Email address: lwyeh@cs.nctu.edu.tw

Publication List

Journal papers

1. L.-W. Yeh, Y.-C. Wang, and Y.-C. Tseng, "iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks", *Int'l J. of Sensor Networks*, Vol. 5, No. 1, 2009, pp. 1-10.

2. M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, and Y.-C. Tseng, "A WSN-based Intelligent Light Control System Considering User Activities and Profiles", *IEEE Sensors Journal*, Vol. 8, No. 10, Oct. 2008, pp. 1710-1721.

3. L.-W. Yeh, C.-Y. Lu, C.-W. Kou, Y.-C. Tseng, and C.-W. Yi, "Autonomous Light Control by Wireless Sensor and Actuator Networks", *IEEE Sensors Journal*, Vol. 10, No. 6, 2010, pp. 1029-1041.

Conference papers

1. Y.-C. Tseng, Y.-C. Wang, and L.-W. Yeh, "iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks", *Int'l Computer Symp.*, 2006.

2. L.-W. Yeh, M.-S. Pan, and Y.-C. Tseng, "Two-Way Beacon Scheduling in ZigBee Tree-Based Wireless Sensor Networks", *IEEE Int'l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2008.

3. M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, and Y.-C. Tseng, "Design and Implementation of a WSN-Based Intelligent Light Control System", *Int'l Workshop on Wireless Ad hoc and Sensor Networks (WWASN)*, 2008.

4. L.-W. Yeh, C.-Y. Lu, Y.-H. Lin, J.-L. Liao, Y.-C. Tseng, C. Chen, and C.-W. Yi, "iLamp: A Sensor-Enhanced Lamp with Surface-Tracking Capability Based on Light Intensity", *IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom)*, 2009.

5. L.-W. Yeh, M.-S. Hsu, Y.-F. Lee, and Y.-C. Tseng, "Indoor Localization: Automatically Constructing Today's Radio Map by iRobot and RFIDs", *IEEE Sensors Conference*, 2009.

Book Chapters

1. Lun-Wu Yeh, Meng-Shiuan Pan, and Yu-Chee Tseng, "ZigBee Wireless Sensor Network and Its Network Formation Problem" (a book chapter in "Handbook on Sensor Networks", World Scientific Publishing. (ISBN: 978-981-283-730-1)

Patents

1. M.-S. Pan, L.-W. Yeh, Y.-C. Tseng, L.-C. Ko, H.-W. Fang, and C.-W. Teng, “Apparatus for a Beacon-enabled Wireless Network, Transmission Time Determining Method, and Computer Readable Medium Thereof”, Taiwan (pending), USA (pending), owned by III.

2. L.-W. Yeh, C.-Y. Lu, Y.-F. Lee, Y.-H. Lin, and Y.-C. Tseng, “Positioning Method and Positioning System Based on Light Intensity”, Taiwan (pending), USA (pending), China (pending), owned by ITRI.

Submitted papers

1. L.-W. Yeh, M.-S. Pan, and Y.-C. Tseng, “Low-Latency Two-Way Beacon Scheduling for Broadcast and Convergecast in ZigBee Tree-Based Networks”, submitted to *Computer Communication*.

2. L.-W. Yeh, M.-H. Hsu, and Y.-C. Tseng, “Design and Implementation of a Self-Guided Indoor Robot Based on a Two-Tier Localization Architecture”, submitted to *Pervasive and Mobile Computing*.