國立交通大學

網路工程研究所

碩 士 論 文

無 線 感 測 網 路 之 自 動 燈 光 控制系統

Autonomous Light Control by Wireless Sensor and Actuator Networks

研 究 生：呂哲彥

指導教授：曾煜棋 教授、易志偉 教授

中 華 民 國 九 十 八 年 六 月

無線感測網路之自動燈光控制系統

Autonomous Light Control by Wireless Sensor and Actuator Networks

研 究 生：呂哲彥 Student：Che-Yen Lu

指導教授：曾煜棋、易志偉 Advisor：Yu-Chee Tseng、

Chih-Wei Yi

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

無線感測網路之自動燈光控制系統

學生：呂哲彥　　　　　　　　　　　　　　　　指導教授：曾煜棋 教授

易志偉 教授

國立交通大學網路工程研究所碩士班

摘　　要

近年來，無線感測網路已經被廣泛的使用在許多的領域中。此篇論文中，我們提出了一個基於無線感測網路之自動燈光控制系統。系統藉由使用者身上配戴燈光感測器之讀數，做為調控燈光的依據。本系統主要針對兩大目標，分別為滿足使用者照明以及節能省電。依據照明範圍的不同，將照明設備分為全區照明以及區域照明設備。且基於全區照明以及區域照明考量因素不同，本論文對於兩種設備分別提出不同的解決方案。對於全區照明，我們提出兩種控制演算法來控制全區照明設備；對於區域照明則提出可追蹤使用者閱讀面之智慧型檯燈。由於不需要額外的定位媒介以及裝置，且能自動根據環境光源的變化作調整，因此我們將本系統稱為「**自動化**」燈光調控系統。本論文中，我們提供實驗數據以及實作雛形結果來驗證系統的可行性。

關鍵字： 智慧型建築、燈光控制、無線通訊、普及運算，無線感測網路以及LED。

Autonomous Light Control by Wireless Sensor and Actuator Networks

student：Che-Yen Lu

Advisors：Prof. Yu-Chee Tseng
Prof. Chih-Wei Yi

Institute of Network Engineering
National Chiao Tung University

ABSTRACT

Recently, wireless sensor and actuator networks (WSANs) have been widely discussed in many applications. In this paper, 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 these results.

Keywords: Intelligent building, light control, pervasive computing, wireless communication, wireless sensor and actuator network, LED.

誌　　　謝

首先，誠摯的感謝曾煜棋教授以及易志偉教授對於我碩士生涯兩年來的指導與鼓勵，並且提供我良好的研究環境以及充足的實驗器材，讓我能夠順利的完成此篇論文並且順利取得碩士學位。

此外，也由衷的感謝葉倫武學長以及廖家良助理。無論在論文研究與寫稿上，葉倫武學長皆細心提供了我許多寶貴的建議以及指導。而在硬體實作以及佈建上，廖家良助理也給予了我許多幫助。另外，我要感謝 HSCC 以及 NOL 實驗室的全體成員，在我碩班兩年中的幫忙與鼓勵。

最後，感謝我的家人以及關心我的人對我的期許以及關懷，使我能夠無虞地完成我的學業。

呂哲彥　於

國立交通大學網路工程研究所碩士班

中華民國九十八年六月

Contents

List of Figures

Chapter 1

Introduction

The rapid progress of wireless communication and embedded MEMS technologies has made *wireless sensor and actuator networks* (WSANs) possible. A WSAN [10][11][20] 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. Applications of WSANs include smart living space [21], localization [13][15], and environmental monitoring [14][23].

Recently, WSANs have been applied to energy conservation applications such as light control [14][16][17][19][22]. Reference [22] uses wireless sensors to control lighting devices according to daylight intensity. Reference [17] defines several user requirements and cost functions. The goal is to adjust lights to minimize the total cost. However, the result is mainly for media production. Considering light control is a trade-off between energy consumption and user satisfaction, reference [19] applies the concept of utility to adjust illuminations so as to maximize the total utility. However, it does not consider the fact that people need different illuminations under different activities. In references [17] and [19], it needs to measure all combinations of dimmer settings and the resulting illuminations at all locations. If there are k interested locations, d dimmer levels, and m lighting devices, the measurement complexity is $O(kdm)$. With pervasive sensors, [16] further reduce the measuring time to $O(km)$. The goal is to satisfy users' demands while optimizing energy efficiency. These works all rely on knowing

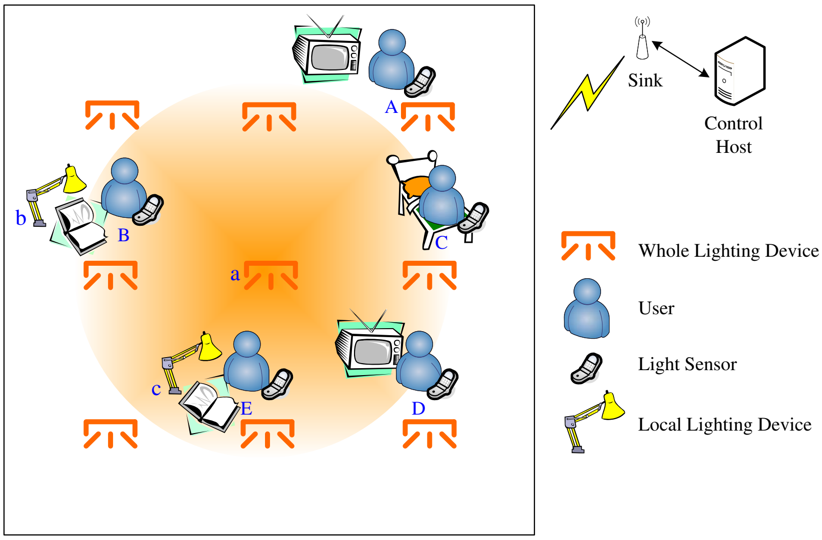


Figure 1.1: The network scenario of our system.

users' current locations, so extra localization mechanisms are needed.

In this work, we propose a light control system that considers users' preferences and energy conservation. Fig. 1.1 shows the network scenario. Each user carries a light sensor and these sensors can help each other to relay their sensing data to the sink node. Then the control host can give commands to lighting devices. We consider LEDs [3][4] serving as *whole* and *local lighting devices*. The former can provide background illuminations for multiple users in wide areas. The latter are similar to desk lamps to provide concentrated illuminations. For example, in Fig. 1.1, device a in the center can provide background illuminations for user B, C and E, and device b can only provide concentrated illumination for user B.

In our system, users may have different illumination requirements according to their activities and profiles. We distinguish from two types of requirements, *background* and *concentrated* ones. For example, in Fig. 1.1, user A is watching television, B is reading a book, and C is sleeping. Both A and B require the same background illuminations, but B needs concentrated illumination, and C requires no background and concentrated illuminations. A user is said to be *satisfied* if the provided background and concentrated illuminations fall into the required ranges. To evaluate the satisfaction level of a user, we further consider a *binary satisfaction* and a *continuous satisfaction models*. The former only returns a satisfaction value of 1 or 0, while the latter returns a value between 0 and 1. 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 follow to local movements of users to provide required illuminations.

The main contributions of this work 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 main lighting sources 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 to track users' current locations.

There rest of this work is organized as follows. Chapter 2 presents the system model. Chapter 3 and Chapter 4 introduce our control algorithms for background and concentrated light sources, respectively. Chapter 5 contains simulation results. Chapter 6 presents our prototyping results. Conclusions are drawn in Chapter 7

Chapter 2

System Model

2.1 Light Measurement Method

In our system, there are n users, $u_1, u_2, ..., u_n$, m whole lighting devices, D_1, D_2, ..., D_m, and m' local lighting devices, $d_1, d_2, ..., d_{m'}$. These devices are all controllable devices. Each user u_i carries a light sensor s_i, which periodically reports its sensed illumination level P_i to the control host. The current luminous intensity emitted by D_i is denoted by C_i^D, and that by d_i is denoted by C_i^d. Considering physical limitations, we assume that C_i^D and C_i^d should satisfy $C_i^{Dmin} \leq C_i^D \leq C_i^{Dmax}$ and $C_i^{dmin} \leq C_i^d \leq C_i^{dmax}$.

We make the following assumptions in our work. First, there exists natural light source, but it may change over time. Second, 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, while allows us to decide the proper intensity of each light source. For local lighting sources, we assume that no such disturbance exists. This allows us to measure the distance between a lighting source and a user. In fact, we even assume that local lighting sources are supported by robot arms and thus they may be moved around to focus to particular places. We will discuss more about this in Chapter 4.

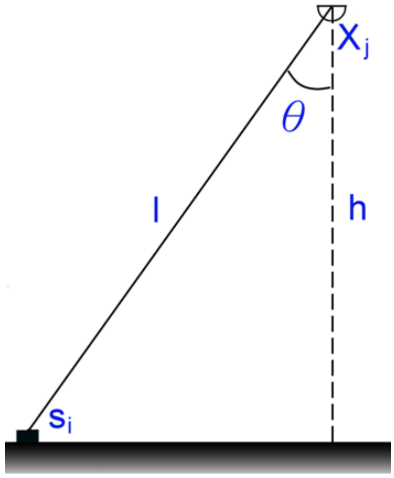


Figure 2.1: Measuring the impact of a light source X_j on a light sensor s_i.

Next, we explain how to model the impact of a light source X_j on a light sensor s_i (refer to in Fig. 2.1). X_j can be a whole lighting source D_j or a local lighting source d_j. Let l and h 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,

$$\Delta L_{i,j} = \frac{\Delta C_j^X \times \cos\theta}{l^2} = \frac{\Delta C_j^X \times h}{l^3}. \tag{2.1}$$

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

$$w_{i,j}^X = \frac{\Delta L_{i,j}}{\Delta C_j^X} = \frac{h}{l^3} \tag{2.2}$$

Intuitively, this implies that even if l and h are unknown, we can still measure $w_{i,j}^X$ 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, when $X_j = D_j$, the impacted is written as $w_{i,j}^D$; when $X_j = d_j$, it is written as $w_{i,j}^d$. The measurement of impact values should be done one-by-one, so the overall complexity is $O(m + m')$. In our work, we will also

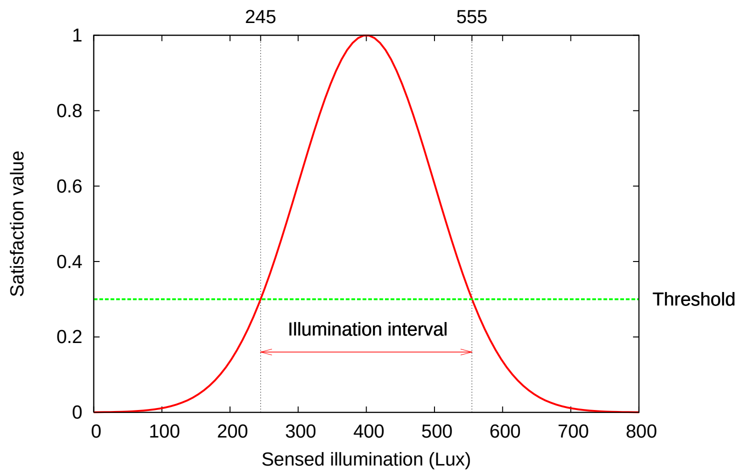


Figure 2.2: An example of continuous satisfaction.

consider only measuring the impacts of some light devices and use interpolation techniques to estimate those unknown impact values to further reduce the measurement cost. In comparison, this is much lower than that of [16], [17] and [19].

Because illuminations are additive [19], the P_i sensed by s_i is the sum of the natural light L_i^{na} and the illuminations provided by whole and local lighting devices

$$P_i \approx \sum_{j=1}^{m} (w_{i,j}^D \times C_i^D) + \sum_{j=1}^{m'} (w_{i,j}^d \times C_i^d) + L_i^{na}. \tag{2.3}$$

P_i can be considered as the concentrated illumination perceived by u_i and the background illumination perceived by u_i can be estimated by $P_i - \sum_{j=1}^{m'} (w_{i,j}^d \times C_i^d)$. In this work, we consider two kinds of user model for background illuminations:

1. *Binary Satisfaction Model*: Each user u_i has a acceptable concentrated illumination interval $[R_i^{cl}, R_i^{cu}]$ and an acceptable background illumination interval $[R_i^{bl}, R_i^{bu}]$. The user is said to be *satisfied* if its concentrated and background illuminations fall within these intervals, respectively.

2. *Continuous Satisfaction Model*: User u_i also has concentrated and back-

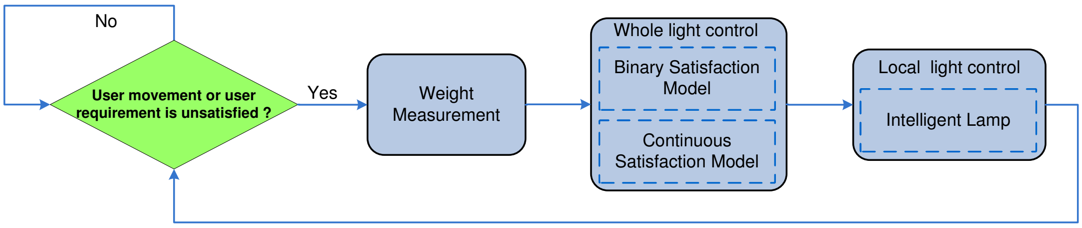


Figure 2.3: Light control flow chart.

ground illumination requirements, but they are specified by utility-like functions as in [19]. The former has a mean μ_i^c, a variance value σ_i^c, and a threshold t_i^c. The latter has a mean μ_i^b, a variance value σ_i^b, and a threshold t_i^b are specified. Given a concentrated illumination x, u_i has a satisfaction value of $f_i^c(x) = exp(\frac{-(x-\mu_i^c)^2}{2(\sigma_i^c)^2})$ and a acceptable concentrated illumination interval $[\mu_i^c - \sigma_i^c\sqrt{-2ln(t_i^c)}, \mu_i^c + \sigma_i^c\sqrt{-2ln(t_i^c)}]$. Similarly, given a background illumination x, u_i has a satisfaction value of $f_i^b(x) = exp(\frac{-(x-\mu_i^b)^2}{2(\sigma_i^b)^2})$ and a acceptable background illumination interval $[\mu_i^b - \sigma_i^b\sqrt{-2ln(t_i^b)}, \mu_i^b + \sigma_i^b\sqrt{-2ln(t_i^b)}]$. Fig. 2.2 shows an example of continuous model with $\mu = 400$, $\sigma = 100$, and $t = 0.3$.

Note that for concentrated illuminations, we assume that it is always possible to meet users' requirements since local lighting devices are very close to users, so no particular model is specified.

2.2 Control Flow

Fig. 2.3 show the light control flow of our system. It is triggered by user movement, periodical check, or inputs from sensors which reflect that some users are not satisfied. The weight measurement block will determine $w_{i,j}^D$ and $w_{i,j}^d$ is discussed in Section 2.1. Then the whole light control and the local light control modules will follow. We will use P_i and $P_i - \sum_{j=1}^{m}(w_{i,j}^d \times C_i^d)$ to measure the concentrate and background illuminations of u_i, respectively, and adjust C_i^Ds and

C_i^ds to achieve our goal. It turns out that decisions of whole or local light control can be made independently of each other.

Chapter 3

Control of Whole Lighting Devices

Under the binary model, we propose to minimize the total energy cost. Under the continuous model, since there is a satisfaction value associated with each user, we propose to maximize users' total satisfaction value.

3.1 Binary Satisfaction Model

Our goal is to determine an amount of adjustment ΔC_i^D on C_i^D for device D_i to meet users' background illumination requirements. Under the binary satisfaction model, we are given the inputs: (1) $C_1^D, C_2^D, \ldots, C_m^D$, (2) $C_1^d, C_2^d, \ldots, C_m^d$, and (3) $P_1, P_2, \ldots, P_n$. Also, from the light measurement method in Section 2.1, we can derive: (1) $w_{1,1}^D, w_{1,2}^D, \ldots, w_{n,m}^D$, (2) $w_{1,1}^d, w_{1,2}^d, ..., w_{n,m'}^d$, and (3) $L_1^{na}, L_2^{na}, \ldots, L_n^{na}$. Our goal is to solve $\Delta C_1^D, \Delta C_2^D, \ldots, \Delta C_m^D$ with the objective function:

$$min \sum_{i=1}^{m} (C_i^D + \Delta C_i^D) \tag{3.1}$$

subject to:

$$R_i^{bl} \leq \sum_{j=1}^{m} w_{i,j}^D \times (C_j^D + \Delta C_j^D) + L_i^{na} \leq R_i^{bu} \quad \text{for all} \quad i = 1 \ldots n \tag{3.2}$$

$$C_i^{Dmin} \leq C_i^D + \Delta C_i^D \leq C_i^{Dmax} \quad \text{for all} \quad i = 1 \ldots m \tag{3.3}$$

Eq. (3.1) is to minimize the total power consumption of whole lighting devices. Eq. (3.2) imposes that all users' background illumination requirements should be met. Eq. (3.3) is to confine the adjustment result within the maximum and the minimum bounds. This is a linear programming problem and can be solved by the Simplex method [9]. However, in reality, there may not exist feasible solutions. In this case, we will gradually relax users' requirements to make this problem feasible. Reference [18] already shows that finding a feasible subsystem of a linear system by eliminating the fewest constraints is NP-hard. 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 $[R_i^{bl} - \alpha, R_i^{bu} + \alpha]$ for each $i = 1...n$, where α is a constant. Then we run the Simplex method again. This is repeated until a solution is found.

3.2 Continuous Satisfaction Model

Under this model, the inputs are: (1) $C_1^D, C_2^D, \ldots, C_m^D$, (2) $C_1^d, C_2^d, \ldots, C_m^d$, and (3) $P_1, P_2, \ldots, P_n$. Again, we can derive: (1) $w_{1,1}^D, w_{1,2}^D, \ldots, w_{n,m}^D$, (2) $w_{1,1}^d, w_{1,2}^d, \ldots, w_{n,m'}^d$, and (3) $L_1^{na}, L_2^{na}, \ldots, L_n^{na}$. The goal is to solve $\Delta C_1^D, \Delta C_2^D, \ldots, \Delta C_m^D$ with the objective function:

$$max \sum_{i=1}^{n} f_i^b\left(\sum_{j=1}^{m} w_{i,j}^D \times (C_j^D + \Delta C_j^D) + L_i^{na}\right) \tag{3.4}$$

subject to:

$$\mu_i^b - \sigma_i^b \sqrt{-2ln(t_i^b)} \le \sum_{j=1}^{m} w_{i,j}^D \times (C_j^D + \Delta C_j^D) + L_i^{na}$$
$$\le \mu_i^b + \sigma_i^b \sqrt{-2ln(t_i^b)} \quad \text{for all} \quad i = 1 \ldots n \tag{3.5}$$
$$C_i^{Dmin} \le C_i^D + \Delta C_i^D \le C_i^{Dmax} \quad \text{for all} \quad i = 1 \ldots m \tag{3.6}$$

Eq. (3.4) is to maximize the sum of satisfaction values of all users. Eq. (3.5) imposes that all users' background illumination requirements should be met. Eq. (3.6) specifies the bounds. This is a non-linear programming problem and can be solved by a sequential quadratic programming (SQP) method [8]. When there is no fea-

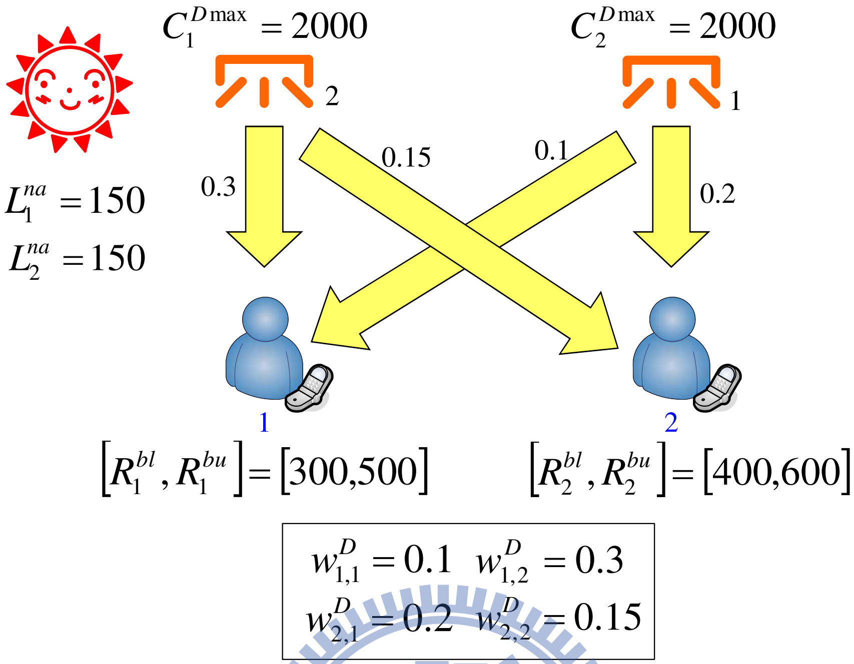


Figure 3.1: An example for the binary satisfaction model.

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

3.3 Examples

For the binary satisfaction model, Fig. 3.1 shows a scenario with users u_1 and u_2, devices D_1 and D_2, and natural light $L_1^{na} = 150$ and $L_2^{na} = 150$. Let $[R_1^{bl}, R_1^{bu}] = [300, 500]$ and $[R_2^{bl}, R_2^{bu}] = [400, 600]$ and the current intensities $C_1^D = 100$ and

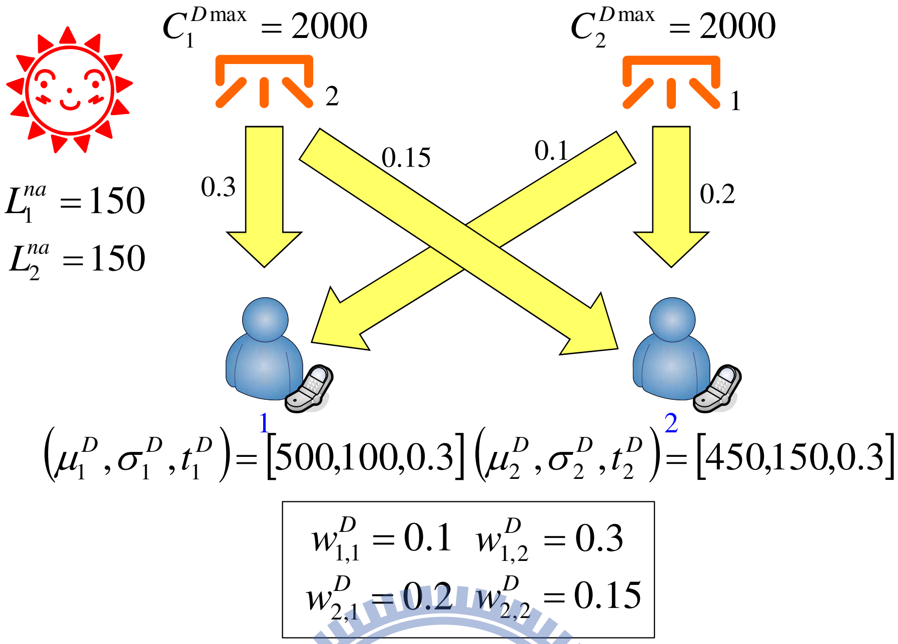


Figure 3.2: An example of continuous satisfaction model.

$C_2^D = 100$. The objective function is:

$$
\begin{aligned}
& \min \quad (\Delta C_1^D + 100) + (\Delta C_2^D + 100) \\
\equiv \quad & \min \quad (\Delta C_1^D + \Delta C_2^D)
\end{aligned}
$$

subject to :

$$
\begin{gathered}
300 \leq 150 + 0.1 \times (100 + \Delta C_1^D) + 0.3 \times (100 + \Delta C_2^D) \leq 500 \\
400 \leq 150 + 0.2 \times (100 + \Delta C_1^D) + 0.15 \times (100 + \Delta C_2^D) \leq 600 \\
0 \leq (100 + \Delta C_1^D) \leq 2000 \\
0 \leq (100 + \Delta C_2^D) \leq 2000
\end{gathered}
$$

Because this problem is feasible, the solution is $\Delta C_1^D = 1066.67$ and $\Delta C_2^D = 11.11$.

For continuous satisfaction model, Fig. 3.2 also shows a scenario with users u_1 and u_2, devices D_1 and D_2, current intensities $C_1^D = 100$ and $C_2^D = 100$, and natural light $L_1^{na} = 150$ and $L_2^{na} = 150$. We assume that $(\mu_1^b, \sigma_1^b, t_1^b) = (500, 100, 0.3)$ and $(\mu_2^D, \sigma_2^D, t_2^b) = (450, 150, 0.3)$ for u_1 and u_2, respectively. Given $t_1^b = 0.3$ and $t_2^b = 0.3$, we can derive $[\mu_1^b - \sigma_1^b\sqrt{-2ln(t_1^b)}, \mu_i^b + \sigma_i^b\sqrt{-2ln(t_i^b)}] = [345, 655]$ 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:

$$
\begin{aligned}
\text{max} \quad & f_1^b\left(150 + 0.1 \times (100 + \Delta C_1^D) + 0.3 \times (100 + \Delta C_2^D)\right) + \\
& f_2^b\left(150 + 0.2 \times (100 + \Delta C_1^D) + 0.15 \times (100 + \Delta C_2^D)\right)
\end{aligned}
$$

subject to:

$$
\begin{gathered}
345 \leq 150 + 0.1 \times (100 + \Delta C_1^D) + 0.3 \times (100 + \Delta C_1^D) \leq 655 \\
167 \leq 150 + 0.2 \times (100 + \Delta C_2^D) + 0.15 \times (100 + \Delta C_2^D) \leq 633 \\
0 \leq (100 + \Delta C_1^D) \leq 2000 \\
0 \leq (100 + \Delta C_2^D) \leq 2000.
\end{gathered}
$$

Again, this problem is also feasible. The solution is $\Delta C_1^D = 255.4$ and $\Delta C_2^D = 966.3$.

Chapter 4

Control of Local Lighting Devices

The above results are able to adjust background illuminations to meet users' needs. In this chapter, we propose a robotic device, called *Intelligent Lamp* (iLamp) to provide concentrated illuminations. Each iLamp has a robot arm with at least four local lighting devices and is supposed to serve one user who has concentrated illumination need at a time. The service scenario is shown in Fig. 4.1. 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. 4.2(a). Note that it is not hard to extend this result to more lighting devices in other geometry mod-

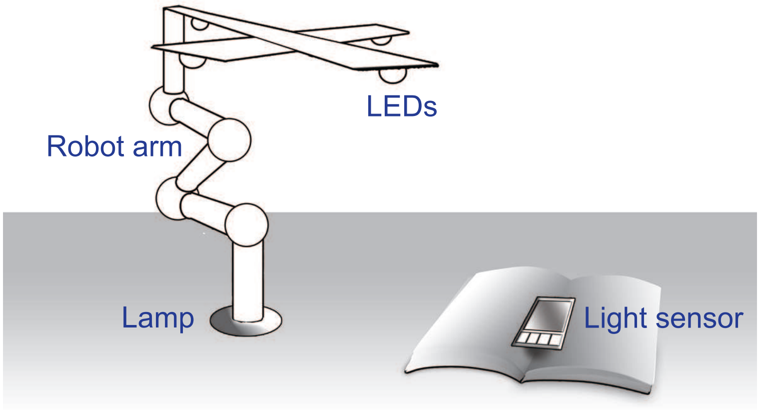


Figure 4.1: Service scenario of an iLamp and a light sensor.

els. Since there is a robot arm, the iLamp should know the coordinate (x_j, y_j, z_j) of d_j, $j = 1 \ldots 4$. 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 as 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 will dissipate 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 will take their middle point as the estimated location of s_i.

1. For each d_j, $j = 1 \ldots 4$, increase its luminous intensity by ΔC_j candela and measure the change of illuminance intensity at s_i, denote by ΔL_{ij}. Accord-

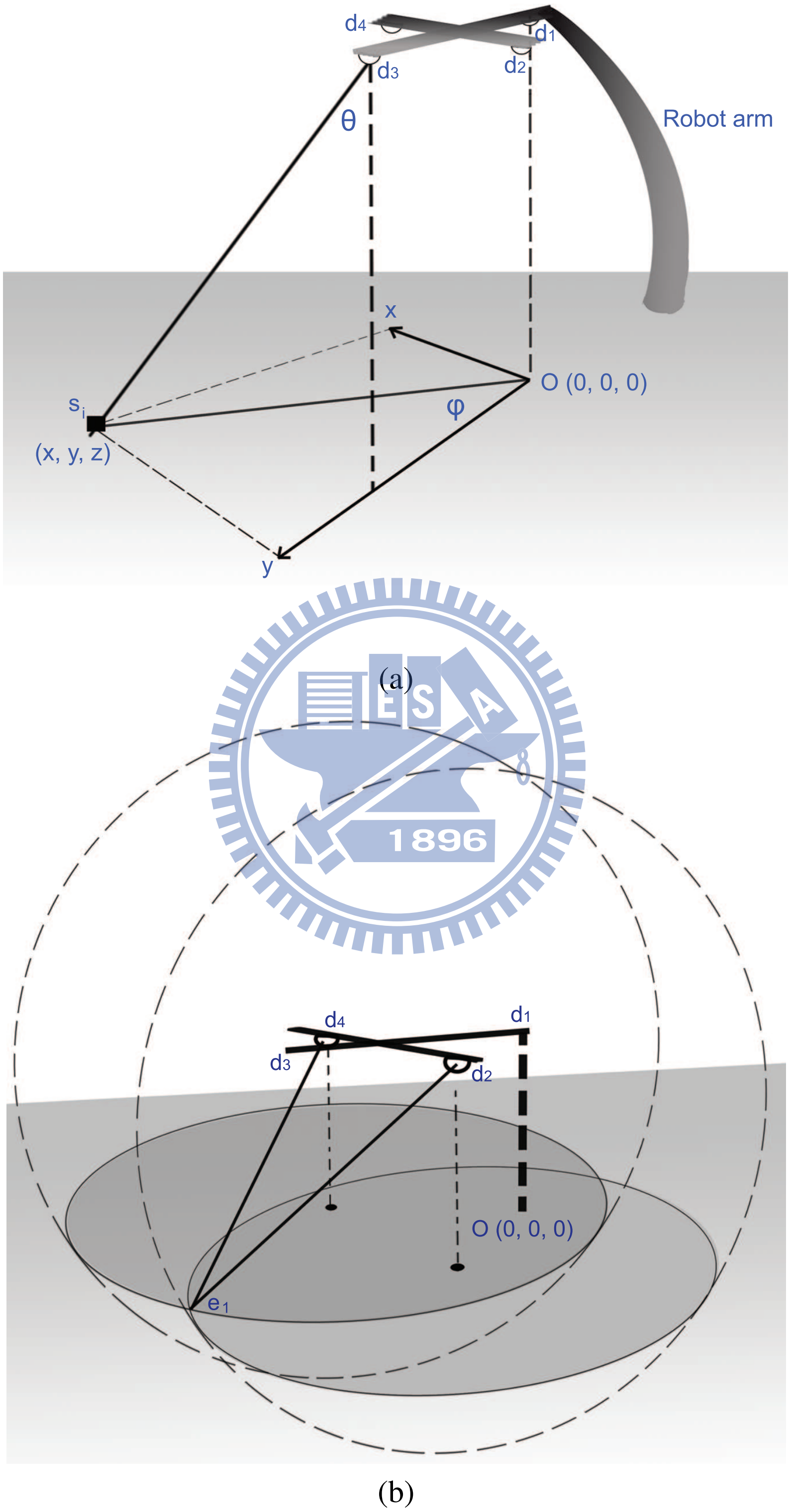


Figure 4.2: The geometry model of iLamp to track the location of s_i.

ing 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

$$\Delta L_{ij} = \frac{\Delta C_i \times z_j}{(\sqrt{(x-x_j)^2+(y-y_j)^2+(z-z_j)^2})^3}. \tag{4.1}$$

2. Observe that the equations for 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 for ΔL_{i1} and ΔL_{i3}, we can pick one point as the estimated location of s_i, called e_1. (Refer to Fig. 4.2(b).)

3. Similarly, the equations for 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, call 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 will move our lighting devices toward the upper side of s_i. This includes two sub-steps. First, we will 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, $j = 1 \dots 4$ to meet the concentrated illumination demand of u_i. From the results in Chapter 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 Chapter 3 can be directly applied again here, so we omit the details.

Chapter 5

Simulation Results

To understand how our schemes for whole lighting control meet users' requirements whole save energy. We have developed a simulator. Two scenarios are considered. Scenario $S1$ is a room of size $10 \times 10m^2$ with 5×5 whole lighting devices. Scenario $S2$ is a room of size $20 \times 20m^2$with 9×9 devices. Both deploy devices as grids. We set all $C_i^{Dmin} = 0$ and all $C_i^{Dmax} = 3000$. We compare with two schemes. The FIX scheme is a very intuitive one assuming that the users' locations are known in advance, we always pick the nearest devices and set them to 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, it picks the nearest device to satisfy the user (if possible). If it still lacks of illumination, the second nearest device is picked to increase its 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 verify both our models.

- Binary Satisfaction Model: We consider two requirement pools, $RP1$ and $RP2$, as shown in Fig. 5.1. Each range R_i in Fig. 5.1 represents an expected illumination interval. A user will randomly one R_i as its requirement. We consider two performance indices here. The first index is the total energy consumption. The second index is called, which reflects the difference between the provided illumination and the required one. The GAP for user u_i

is:

$$GAP(u_i) = \begin{cases} 0 & \text{if } R_i^{bl} \leq P_i \leq R_i^{bu} \\ \min(|R_i^{bl} - P_i|, |R_i^{bu} - P_i|) & \text{otherwise} \end{cases} \tag{5.1}$$

We will measure the average GAP of all users.

Fig. 5.3 and Fig. 5.4 show our simulation results under different combinations of $S1/S2$ and $RP1/RP2$. In Fig. 5.3 (a), we see that our scheme is most energy-efficient while keeps the average GAP close to zero. This is because the requirement intervals in $RP1$ have common overlapping, which allows our system to satisfy all users in most cases. Note that although FIX-1000 uses less energy, its GAP is much larger. Fig. 5.3 (b) adopts $RP2$. Because some requirements are violated, our scheme also induces some gaps. However, our scheme is most energy-efficient. Fig. 5.4 considers $S2$ 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. 5.2. 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. 5.5 and Fig. 5.6 show out 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.

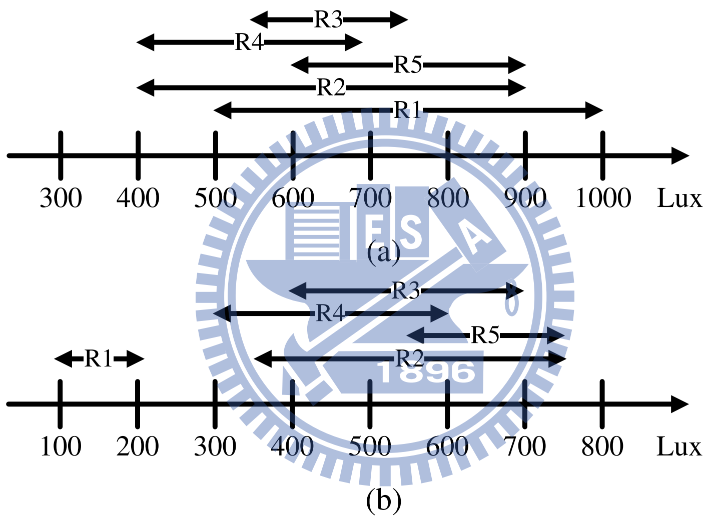
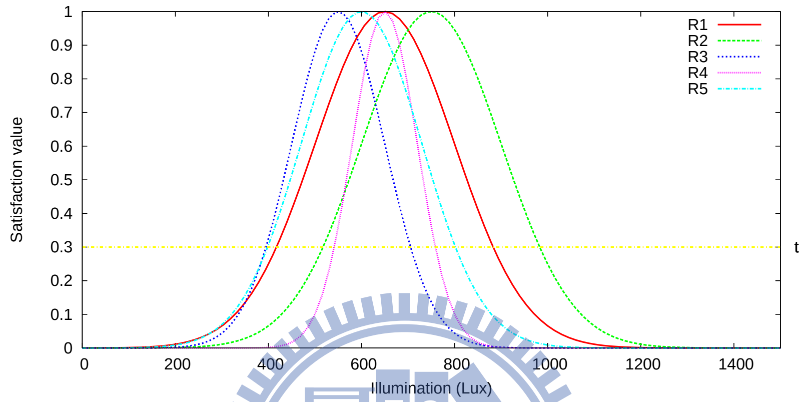
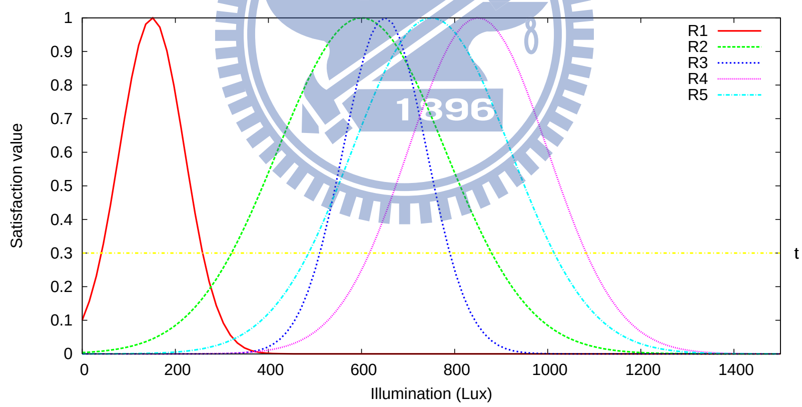


Figure 5.1: Requirement pools: (a) $RP1$ and (b) $RP2$.



(a)



(b)

Figure 5.2: Requirement pools: (a) $RP3$ and (b) $RP4$.

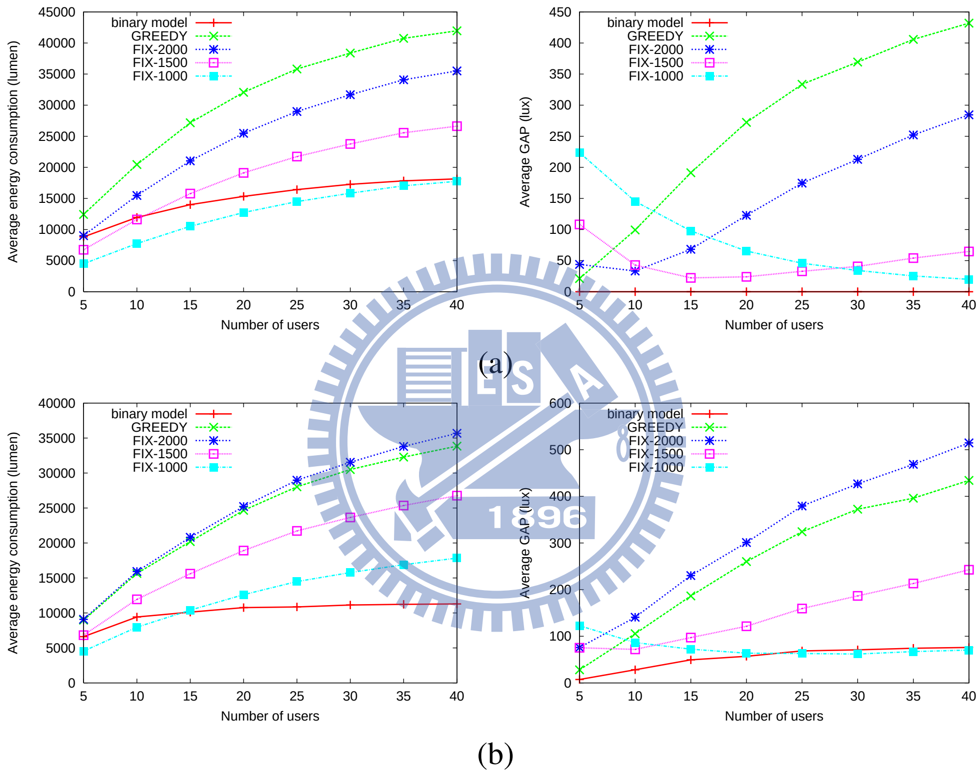
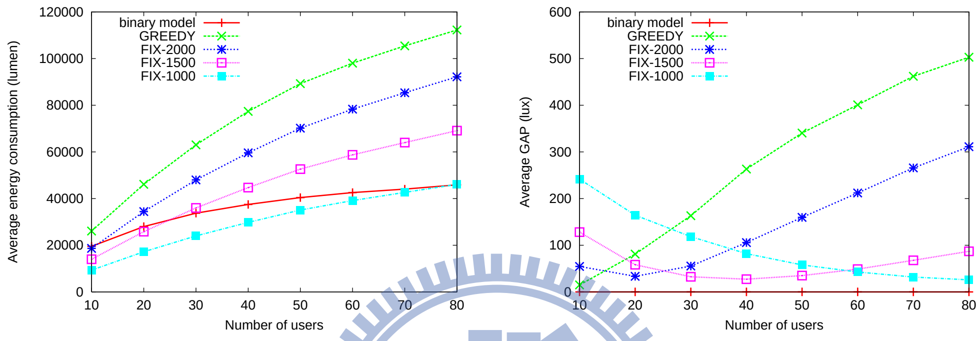
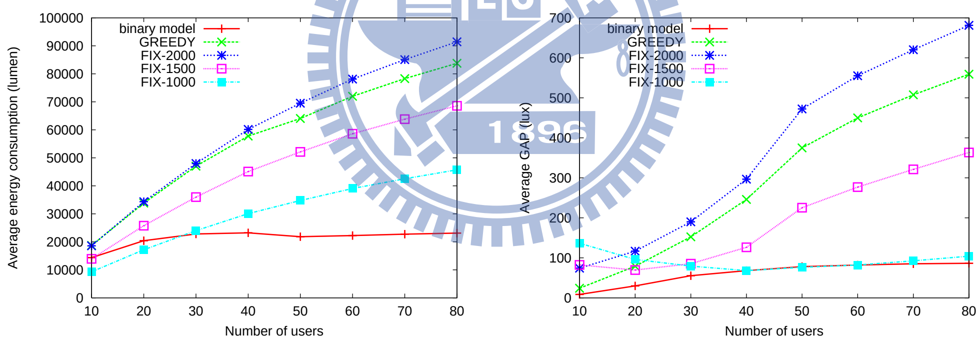


Figure 5.3: Comparison under the binary satisfaction model: (a) network scenario $S1$ and pool $RP1$ and (b) network scenario $S1$ and pool $RP2$.

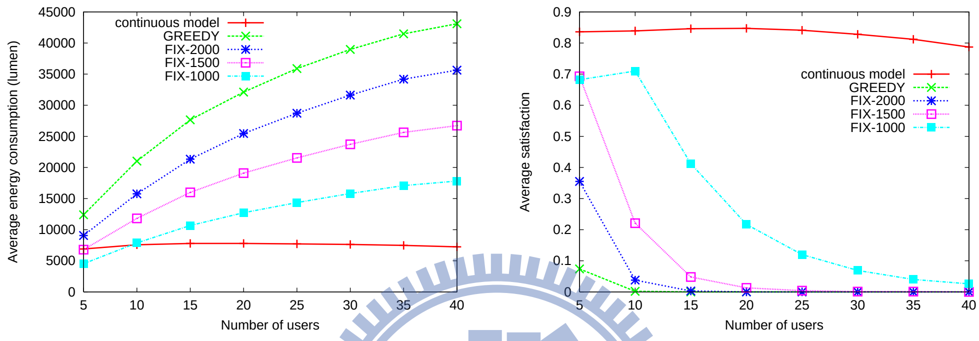


(a)

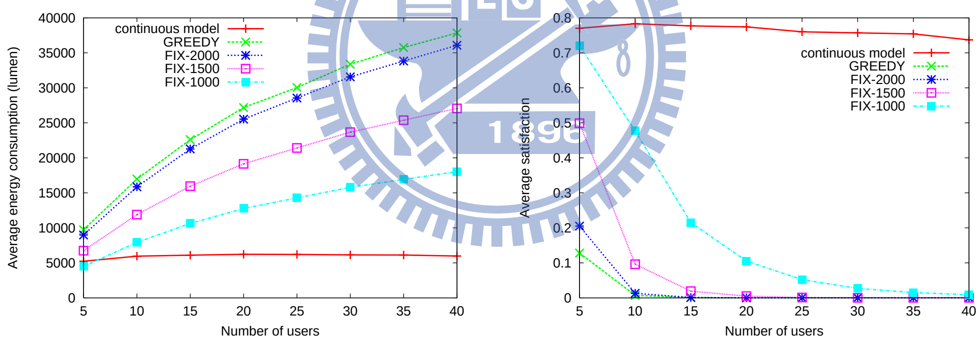


(b)

Figure 5.4: Comparison under the binary satisfaction model: (a) network scenario $S2$ and pool $RP1$ and (b) network scenario $S2$ and pool $RP2$.



(a)



(b)

Figure 5.5: Comparison under the continuous satisfaction model: (a) network scenario $S1$ and pool $RP3$ and (b) network scenario $S1$ and pool $RP4$.

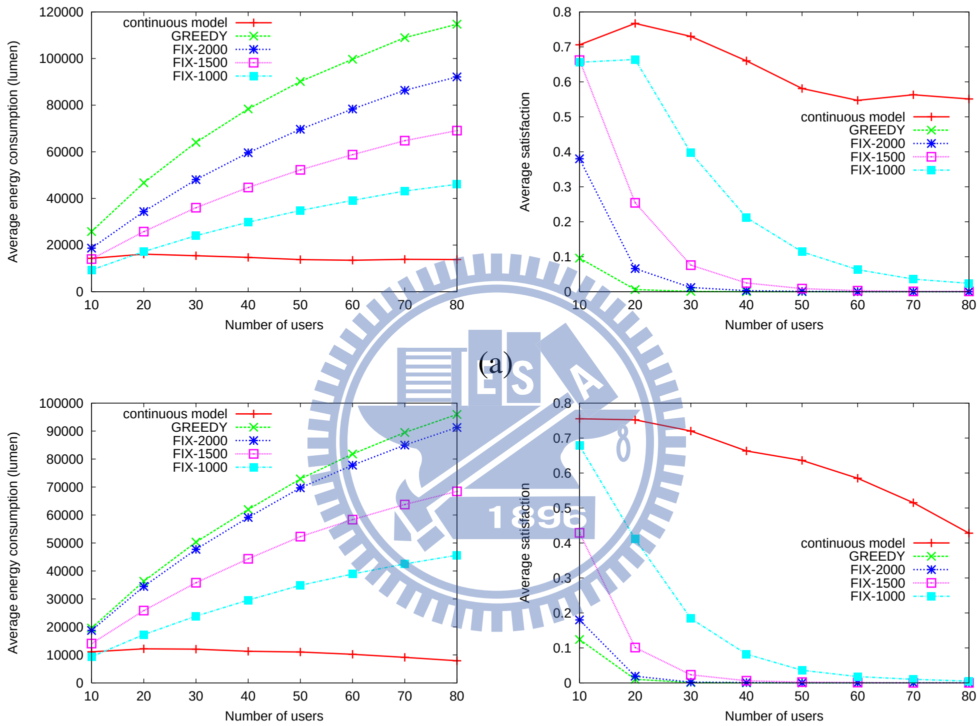


Figure 5.6: Comparison under the continuous satisfaction model: (a) network scenario $S2$ and pool $RP3$ and (b) network scenario $S2$ and pool $RP4$.

Chapter 6

Prototyping Results

We have developed a prototype to verify our results. Fig. 6.1 shows the system architecture. User can carry a badge with a light sensor. User's preference can be configured via the badge. Then the control host can make decisions and send them to lighting devices. We test our system in a room of size $4 \times 4\ m^2$ with 4×5 whole lighting devices. Below, we introduce each device, followed by our testing results.

6.1 User Badge and Light Sensor

The user badge has a wireless module Jennic (JN5139) [2], a TFT LCD ILI9221 panel [7], some buttons as input devices, and a light sensor TSL230 [5]. JN5139 is a single-chip microprocessor with an IEEE 802.15.4 [12] module. The front side, back side, and graphic user interface (GUI) are shown in Fig. 6.2. The outlook of a badge is like a bookmark. User can specify their preference via our GUI and buttons.

6.2 Whole Lighting Device

We use LEDs as light sources. Whole lighting devices are deployed as a $4m \times 5m$ grid on the ceiling. 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

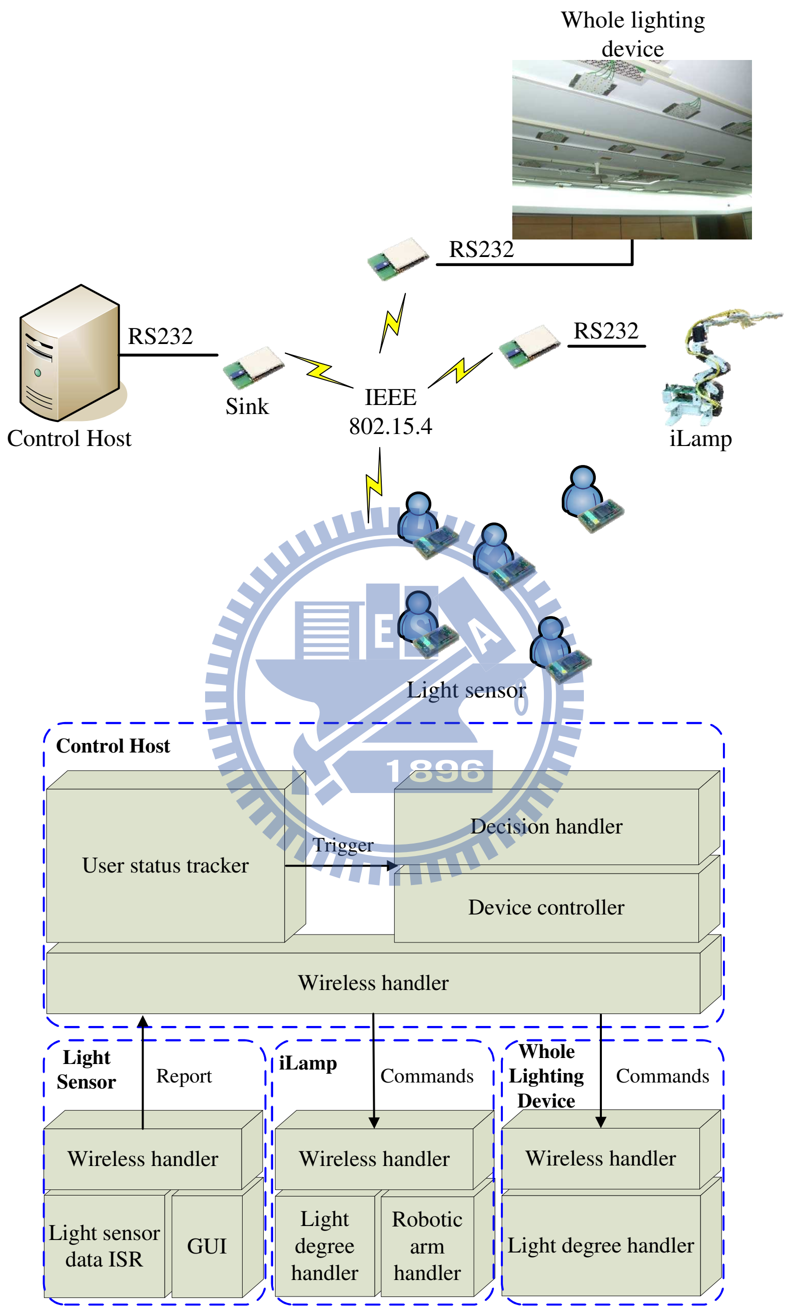


Figure 6.1: Hardware and software system architecture of our prototype.

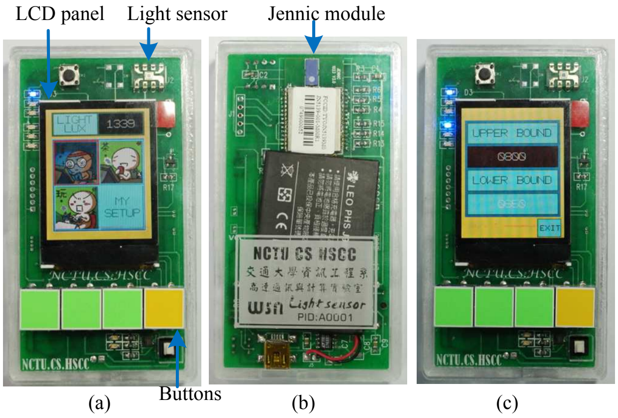


Figure 6.2: User badge, which looks like a bookmark.

Figure 6.3: Testing environment and a whole lighting device.

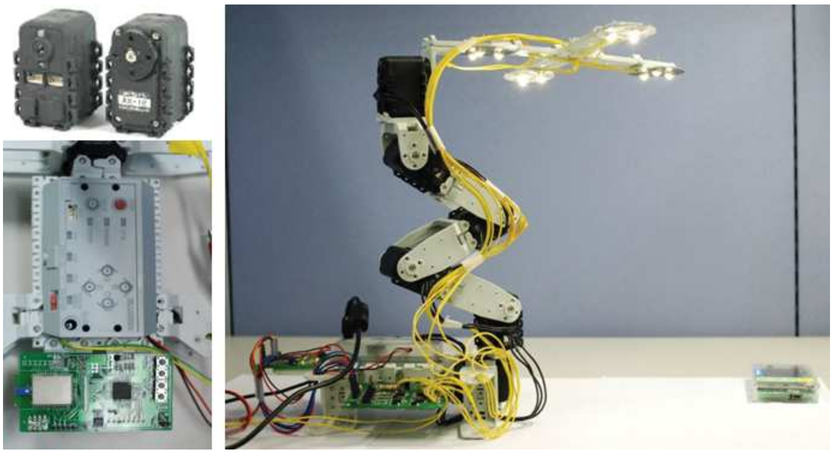


Figure 6.4: The demonstration of iLamp.

modulation of digital input/output (DIO) to control luminous intensity of light sources. Each LED has 20 levels, ranging from 0% to 100% luminous intensity. Fig. 6.3 shows our prototype.

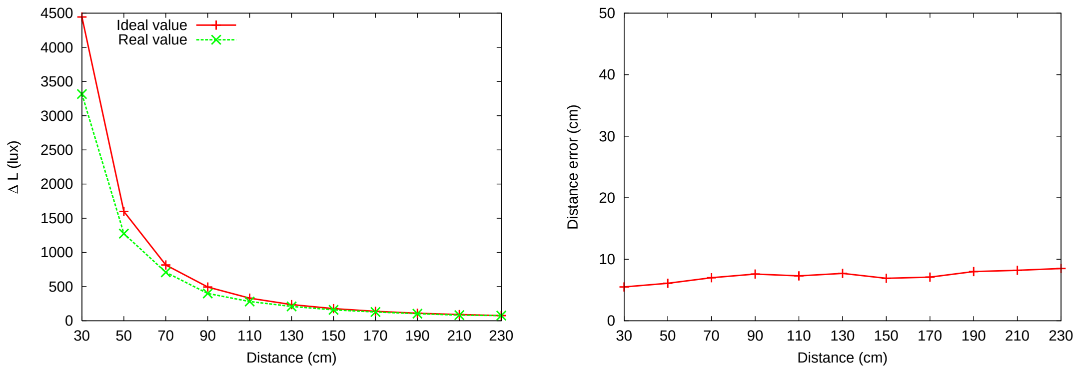
6.3 iLamp

Fig. 6.4 shows the iLamp, which a robot arm, four sets of LEDs, and a JN5139 module. The robot arm consists of six Dynamical AX-12 actuators [1] as the lamp holder. Each AX-12 actuator can rotate from $0°$ to $300°$ at accuracy of $0.33°$. LEDs are the same as whole lighting devices.

6.4 Control Host

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

- User Status Tracker: This component checks current illuminations of all users periodically and, when needed, updates users' requirements. If it de-



(a) (b)

Figure 6.5: Comparison between ideal and real value with fixed candela.

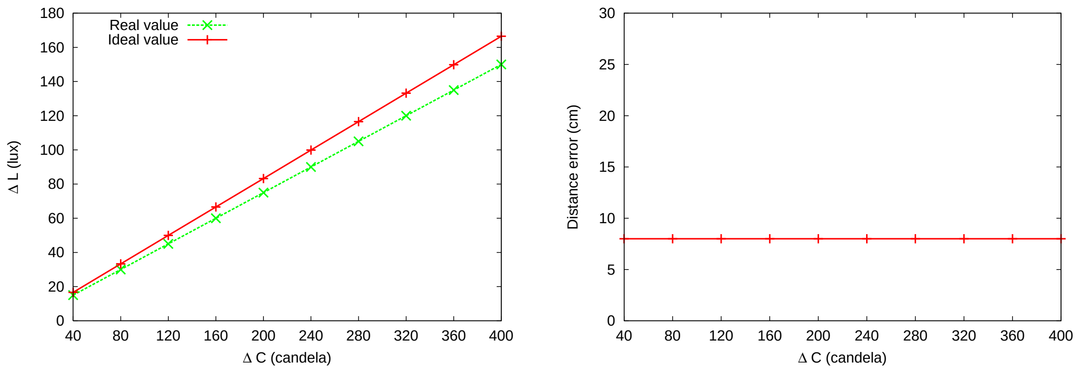
tects that a user's requirement is not satisfied or is updated, a trigger will be sent to the Decision Handler.

- Decision Handler: This component realizes our control algorithms. It is triggered by the User Status Tracker. The linear and non-linear programming are resolved and translated by MATLAB to a JAVA program [6]. The results to Device Controller to adjust lighting devices

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

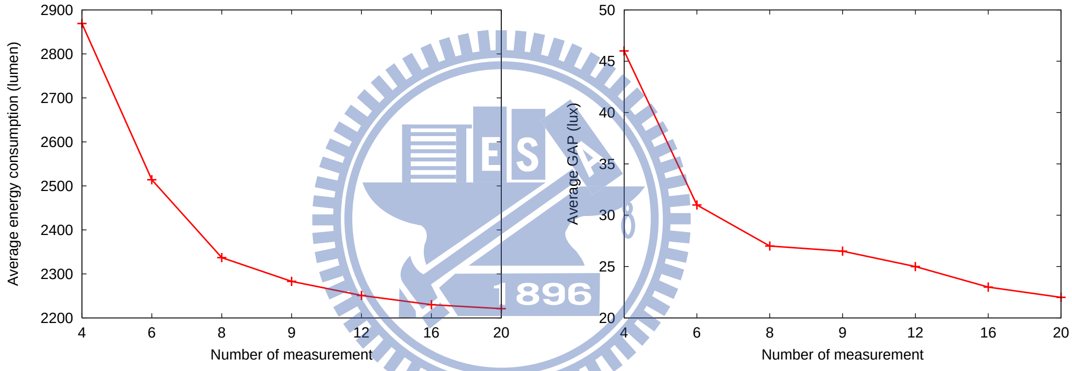
6.5 Performance Verification

In this section, we measure the effectiveness of our model. We verify the correctness of Eq. (2.1) through varying the distance between light sensor and LED or the candela of LED. The real values (i.e., experimental results) and ideal values (i.e., calculating results) are shown in Fig. 6.5 and Fig. 6.6. In Fig. 6.5(a), we fix the candela (ΔC) of LED and set distance from 30 to 230 cm to measure received illumination (ΔL) from light sensor. According to the results of Fig. 6.5(a), we can calculate the difference between ideal and real distance in Fig. 6.5(b). simi-



(a) (b)

Figure 6.6: Comparison between ideal and real value with fixed distance.

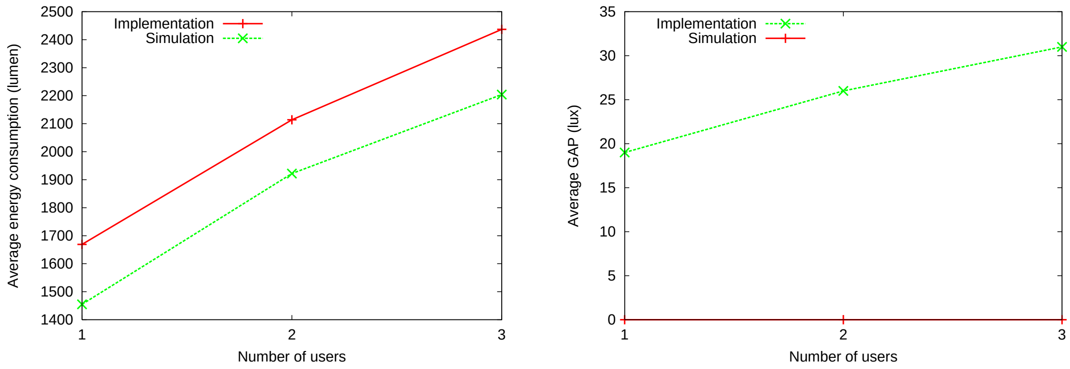


(a) (b)

Figure 6.7: Interpolation under scenario of $RP1$ and 3 users.

larly, in Fig. 6.6(a), we fix the distance between light sensor and LED and set the candela of LED (ΔC) from 40 to 400 candela to measure the received illumination (ΔL) from light sensor. According to the results of Fig. 6.6(a), we can calculate the difference between ideal and real distance in Fig. 6.6(b). In Fig. 6.5(a) the difference of ideal value and real value are almost the same when distance is over 70 cm. In Fig. 6.5(b) and Fig. 6.6(b), we see that all of distance errors are quite small (less than 10 cm).

As shown in Fig. 6.7, we also measure the effectiveness of interpolation. Interpolation is a trade-off between measuring time and energy-saving. Three users



(a) (b)

Figure 6.8: Comparison between implementation and simulation results.

are in this environment and we randomly choose three requirements in $RP1$. Fig. 6.7(a) and Fig. 6.7(b) show that when the number of measurement point increase, average energy consumption and average GAP are also decrease. Also, Fig. 6.8 shows the implementation against simulation results under scenario of $RP1$.

Chapter 7

Conclusions

In this work, 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 surface-tracking scheme is proposed. Our system can dynamically adapt to environment changes and do not need to track users' current locations. Also, we show that our system can be implemented into a real-time system which is different from other light control system.

Besides of illumination, there are lots of factors people concern about. In this work, we only discuss how to control lights in an indoor environment. Because characteristics of all factors are different, we can not directly apply our system to other environmental factors, such as sound, temperature and humidity. Hence, in the future, we may design a system which extend to other factors.

Bibliography

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

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

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

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

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

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

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

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

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to algorithms. In *Cambridge, MA: MIT Press*, 2001.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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):1710–1721, 2008.

[17] H. Park, M. B. Srivastava, and J. Burke. Design and implementation of a wireless sensor network for intelligent light control. In *Proc. on ACM IEEE Int. Conference on Information processing in sensor networks*, 2007.

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

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.