University of California

Merced

Usage Based Building Management through Wireless Sensor Networks

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering and Computer Science

by

Varick L. Erickson

2014

The dissertation of Varick L. Erickson is approved.

David E. Culler

Miguel Á. Carreira-Perpiñán

Ming-Hsuan Yang

Alberto E. Cerpa, Committee Chair

University of California, Merced

2014

Dedicated to Edward, Han, and Shanna Erickson.

Table of Contents

1 Introduction **1**

1.1 Goals 1

1.1.1 Usage based Energy Savings 1

1.1.2 Quality of Service 2

1.2 Contributions 3

1.3 Motivation and Overview of Concepts 4

1.3.1 Occupancy Monitoring 4

1.3.2 Occupancy Prediction 4

1.3.3 HVAC Control 5

1.3.4 Occupant Feedback 5

1.4 Organization 5

2 Related Work **8**

2.1 Usage based Conditioning 8

2.1.1 Occupancy Monitoring 9

2.1.2 Occupancy Prediction 18

2.1.3 Static Occupancy Models 19

2.1.4 Time Series Forecasting 20

2.1.5 Markov Models 24

2.1.6 Classification Based Prediction 27

2.1.7 Predictive Conditioning 28

2.2 Quality of Service 30

2.2.1 PMV Measurement and Actuation 30

2.2.2 Participatory Sensing 33

3 Occupancy Modeling 36

3.1 Data Collection 37

3.2 Multivariate Gaussian Model 38

3.3 Markov Chain Model 41

3.3.1 Closest Distance Markov Chain 44

3.3.2 Moving Window Markov Chain 46

3.3.3 Blended Markov Chain 48

3.3.4 Scalability to Large Buildings 50

3.4 Markov Chain Performance 52

3.4.1 Comparison Metrics 52

3.4.2 Evaluation 53

3.5 Summary 57

4 OPTNet: OPtical Turnstile Network 59

4.1 Inferring Occupancy 59

4.2 Motivation 60

4.3 Overview 61

4.4 OPTNet 61

4.4.1 Overview and Design Challenges 63

4.4.2 Camera Placement 64

4.4.3 Transition Detection 66

4.4.4 OPTNet Evaluation 70

4.5 BONet 76

4.6 Particle Filter 76

4.6.1 Particle Filter Results 81

4.7 Summary 83

5 ThermoSense: Thermal Based Occupancy Sensing 85

5.1 ThermoSense 86

5.1.1 Hardware 88

5.1.2 Power Consumption 89

5.2 Occupancy Classification 90

5.2.1 PIR Sensor Input 91

5.2.2 Thermal Background 93

5.2.3 Feature Vectors 94

5.2.4 K-Nearest Neighbors 96

5.2.5 Linear Regression 97

5.2.6 Artificial Neural Network 98

5.2.7 Filter 98

5.2.8 ThermoSense Performance 98

5.2.9 Model Discussion 100

5.3 Summary 101

6 OBSERVE: Occupancy-Based System for Efficient Reduction of hVac Energy 103

6.1 HVAC Conditioning Strategy 104

6.1.1 Conditioning Criteria 104

6.2 OBSERVE Temperature Control Strategy 106
6.3 Evaluation 108
6.4 Performance Results 108
6.4.1 Building Energy Simulator 108
6.4.2 Energy Savings 111
6.4.3 Conditioning Effectiveness 115
6.5 Conclusions 119
7 POEM: Power-efficient Occupancy-based Energy Management System 120
7.1 Overview 121
7.2 Actuation Scheduler Interface 121
7.2.1 Actuation Algorithm 123
7.3 POEM System Evaluation 125
7.3.1 Energy Savings 127
7.3.2 Live Deployment Results 129
7.3.3 Calibrated Model Results 130
7.3.4 Conditioning Effectiveness 132
7.4 Return on Investment Analysis 135
7.5 Lessons Learned 138
7.6 Summary 140
8 OBSERVE: ThermoSense 141
8.1 Energy Analysis 142
8.2 Conditioning Effectiveness 144

8.3 Summary 146

9 Thermovote 148

9.1 Thermovote Overview 151

9.1.1 Design Considerations 151

9.1.2 Systems 153

9.2 Crowd Driven Control 155

9.2.1 Baseline Control 156

9.2.2 Learned Control Schedule 158

9.2.3 Real-Time Control 161

9.3 User Studies 161

9.4 User Studies 164

9.4.1 Surroundings 164

9.4.2 Thermal Comfort: Studies 1-3 166

9.4.3 Thermal Comfort: Long Term Studies 169

9.4.4 Vote Consistency 174

9.4.5 Maintenance and Management 177

9.4.6 Energy Consumption 178

9.5 Thermal Comfort Models 180

9.5.1 Model Development 182

9.5.2 EnergyPlus Simulated AMV 183

9.5.3 EnergyPlus Simulated Input Bias 186

9.5.4 Voting Scale Comparison 187

9.6 Discussion 189

9.6.1 User Participation 189

9.6.2 Voting Bias 190

9.6.3 User Psychology 190

9.6.4 System Issues 191

9.6.5 Limitations 191

9.7 Summary 192

10 Conclusion and Future Work 194

References 197

List of Figures

1.1 Concept Relationships 7

3.1 The left shows the ten areas data was collected for and the 18 boundaries (gray lines) defining the areas. The right shows a graph representation. 37

3.2 5 days of ground truth occupancy data for eight areas. 39

3.3 The occupancy state representation. 41

3.4 Example of the closest distance metric with state $(Hall, Rm1, Rm2, Rm3)$ and distance 5. 45

3.5 Relationship of blending coefficient, hourly time slot, and transition matrices. 49

3.6 Comparison of training data with the models for several areas. . . 51

3.7 The difference of the average static occupancy duration from 7am - 10pm. 54

3.8 Each boxplot is for the JSDs observed within the time slot where the + is an outlier. 55

3.9 Flows for Hall 1, Office 1, and Office 2 and the min and max flows observed in the testing set. 56

4.1 System components 62

4.2 Transition areas (bigger blue boxes) and trigger areas (small red boxes). 62

4.3 Above is a summary of transition classification procedure. Previous image background subtraction detects movement within a transition area. The image sequence is then concatenated to form a larger matrix shown as a picture on the right of Step 2. A feature vector is then constructed where each position corresponds to the number of active pixels in each column of the matrix (left Step 2). This vector is then classified using KNN against known labeled data (Step 3). 65

4.4 The sequence of the trigger state helps indicate direction. The red box indicates the trigger state is active. 68

4.5 Overall accuracy over 24 hours as a function of training set size. . 71

4.6 Ground Truth compared with KNN model for two different transition areas. 72

4.7 Occupancy error over time with/without maximum/minimum occupancy limits. 74

4.8 A single transition error continues to affect occupancy estimate. . 74

4.9 PIR node deployed on ceiling 76

4.10 PIR accuracy compared with ground truth. 77

4.11 Sensor output error distribution. 79

4.12 Occupancy over time for the transition model, the sensor output and the particle filter for 2 areas 80

4.13 Occupancy error over 2 weeks. 82

5.1 Grid-Eye attached to a Tmote (left). Enclosure containing both the Grid-Eye and PIR (right). 87

5.2 8x8 thermal array sensing an occupant. 88

5.3 Energy usage for different duty cycles. 89

5.4 Occupancy Classification Process 91

5.5 PIR compared to ground truth. 92

5.6 PIR evaluation of all three rooms 92

5.7 Plot of residual distribution 99

5.8 Model raw and filtered outputs. 99

5.9 NRMSE of a ThermoSense node in a zone. 101

6.1 Main building parameters used. 109

6.2 The left shows the ten areas data was collected for and the 18 boundaries (gray lines) defining the areas. 110

6.3 Simulated SCOPES data. 111

6.4 The breakdown of the energy consumption for each month and strategy for three different locations. 112

6.5 Improvement of energy consumption over baseline. 113

6.6 Monthly RMSE temperature for each strategy for Fresno. 116

6.7 Total building ventilation rates for each strategy for one particular summer day. 118

7.1 Architecture of the POEM system. 122

7.2 Energy consumption breakdown for 7 representative days. PIR and camera/PIR based POEM saves on average 21.1% to 26.0% respectively. 128

7.3 Supply air (SA) flows through coils, heating or cooling the air. . . 128

7.4 Energy consumption based on calibrated energy simulation. On average PIR and Camera based strategies save 24.5% and 31.0% annually respectively. 130

7.5 RMSE (F^o) between the target and observed room temperatures . 133

7.6 RMSE (F^o) between the target and observed room temperatures from the calibrated model. 134

7.7 Ventilation and CO_2 levels 136

7.8 The left compares savings for different energy prices. The right shows savings over 80 year period. 139

8.1 Monthly heating and cooling consumption for the different strategies.143

8.2 The temperature RMSE for the periods when the room was occupied during the summer. 144

8.3 Ventilation effectiveness of the strategies. 145

9.1 The different methods of input provided. 152

9.2 Variance of user opinion of what constitutes a comfortable temperature. 152

9.3 Architecture of the Thermovote system. Here we show both realtime and learned versions of the system. 154

9.4 Layout of offices and labs. 163

9.5 The top figure shows the initial estimate PMV estimate for this area, which differs greatly from the user feedback. Based on this initial PMV estimate and user feedback, we can correct the PMV estimate. The middle figure shows when we condition according to the corrected PMV estimate. Though people still indicate they are cold, the frequency of the vote is reduced. The bottom figure shows the system running in real time. The frequency of the vote is reduced dramatically indicating that occupants are comfortable. The corrected PMV is actually higher than 0, indicating that our original correction produces temperatures that are slightly cool. . 165

9.6 Airflow experienced by users. 166

9.7 Study results over the course of 5 weeks. 168

9.8 The weekly vote average and temperature change for different areas. 172

9.9 Effect of activating and deactivating Thermovote. 174

9.10 Variation of votes for zones with multiple occupants. The dark thick line is median. 174

9.11 Vote bias for areas occupied by multiple/single occupants. 176

9.12 Comparison of the original data and KDE. 181

9.13 Energy above baseline. 186

9.14 Simulated AMV Distributions. 186

9.15 The convergence of the different scales for rooms with multiple occupants. 188

List of Tables

5.1 Energy Usage for independent components. 89

5.2 Parameters of linear model and fit metrics. 97

5.3 Evaluation of the models used. 101

7.1 Zone information for deployment. The ramping duration is the time required to reach the target temperature from the setback temperature. 126

7.2 Prices of camera and PIR nodes 138

9.1 Parameters used for initial PMV estimate and temperature set-points.158

9.2 Demographics of the users. 162

9.3 Proximity to window or exterior wall. 164

9.4 Survey results for studies 1, 2, and 3. 167

9.5 Long term comfort and satisfaction. 170

9.6 User interaction with facilities management before Thermovote. . 178

9.7 Correlation coefficient of thermal comfort with several parameters. 182

9.8 Point scale results, multiple occupants. 187

9.9 Point scale results, single occupant. 190

Acknowledgments

I would like to thank my Alberto Cerpa for his support and advice throughout the last six years. I would also like to thank my committee members, Miguel Á. Carreira-Perpiñán, Ming-Hsuan Yang, and David E. Culler for their advice and suggestions. Finally, I am also thankful for the support of my past and current colleagues, Ankur Kamthe, Lun Jiang, Tao Liu, Stefan Achleitner, Alex Beltran, Daniel Winkler, Niloofar Piroozi, Andreas Kolling, Benjamin Balaguer, Gorkem Erinc, Chao Qin, Weiran Wang, Max Vladymyrov and Mohsen Farhadloo.

Vita

2001–2006	B.S. Mathematics and Computational Science, Stanford University.
2003–2006	M.S. Statistics, Stanford University.
2006–2008	Teacher, Mathematics, Ceres Unified School District
2008–2011	Teaching Assistant, Electrical Engineering and Computer Science Department, UC Merced
2011–Present	Research Assistant, Electrical Engineering and Computer Science Department, UC Merced
2012–Present	Campus Energy Manager, Facilities, UC Merced

Publications

Varick L. Erickson, Miguel A. Carreira-Perpinan, Alberto E. Cerpa, ”Occupancy Modeling and Prediction for Building Energy Management,” ACM Transactions on Sensor Networks (TOSN), August, 2014.

Alex Beltran, Varick Erickson (Co-primary Author), Alberto E. Cerpa, ThermoSense: Occupancy Thermal Based Sensing for HVAC Control, In Proceedings of the Fifth ACM Workshop on Embedded Sensing Systems for Energy-Efficient Buildings (Buildsys 2013), November, 2013.

Varick Erickson, Alex Beltran, Daniel A. Winkler, Niloufar P. Esfahani, John

Lusby, Alberto E. Cerpa, ThermoSense: Thermal Array Sensor Networks in Building Management, In Proceedings of the Fifth ACM Workshop on Embedded Sensing Systems for Energy-Efficient Buildings (Buildsys 2013), November, 2013.

Varick Erickson, Stefan Achleitner, Alberto E. Cerpa, POEM: Power-efficient Occupancy-based Energy Management System, In Proceedings of the Twelfth ACM/IEEE International Conference on Information Processing on Sensor Networks (IPSN 2013), April, 2013.

Varick Erickson, Alberto E. Cerpa, Thermovote: Participatory Sensing for Efficient Building HVAC Conditioning(Link) Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficient Buildings (Buildsys 2012), November, 2012.

Varick Erickson, Miguel A. Carreira-Perpinan, Alberto E. Cerpa, OBSERVE: Occupancy-based system for efficient reduction of HVAC energy, In Proceedings of the Tenth ACM/IEEE International Conference on Information Processing on Sensor Networks (IPSN 2011), April, 2011.

Varick Erickson, Alberto E. Cerpa, Occupancy based demand response HVAC control strategy, In Proceedings of the Second ACM Workshop on Embedded Sensing Systems for Energy-Efficient Buildings (Buildsys 2010), November, 2010.

Varick Erickson, Yiqing Lin, Ankur Kamthe, Rohini Brahme, Alberto E. Cerpa, Michael D. Sohn, Satish Narayanan, Energy Efficient Building Environment Control Strategies Using Real-time Occupancy Measurements, In Proceedings of the 1st ACM Workshop On Embedded Sensing Systems For Energy-Efficient Buildings (Buildsys 2009), November, 2009

Abstract of the Dissertation

Usage Based Building Management through Wireless Sensor Networks

by

Varick L. Erickson

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Merced, 2014

Professor Alberto E. Cerpa, Chair

Wireless sensor networks (WSNs) is a field with broad variety of applications. Its flexibility for remote continuous measurement lends itself to applications ranging from locating snipers to measuring volcanic activity. One application that stands to substantially benefit from WSNs is building management. Buildings currently account for 41% of the total energy consumption of U.S. [bed11]. Reducing this energy is of critical importance if we are to achieve sustainability. In most commercial buildings, many rooms remain unoccupied or are conditioned assuming maximum occupancy. By relaxing temperature setbacks and adjusting ventilation to match actual occupancy, significant energy savings are possible. This Dissertation examines the use of wireless sensor networks for the purpose of building energy management and actuation. It explores the design and development of wireless sensor networks for building energy management, how data from these deployments are utilized, the development and implementation of data driven occupancy models to perform simulation and prediction, how data models are used to actuate building management systems, and how crowd-sourced data can be integrated into building control strategies. We show based on real-world data that 30% energy savings is possible through usage based strategies and that 80% occupant satisfaction rates are possible by occupant driven control strategies.

CHAPTER 1

Introduction

For the past two decades, buildings have consumed a significant portion of the energy used in the United States. In 2008, buildings accounted for 41% of the total energy consumption of U.S. [bed11]. Heating, ventilation, and air-conditioning (HVAC) along with lighting account for approximately 50% of the total building energy usage [bed11]. At the same time, the total amount of energy being consumed has also increased. From 1980 to 2008, the total amount of energy consumed by buildings increased 51%. Over the years, lowering this usage has become increasingly critical. The main thesis of this dissertation is the development and evaluation of wireless network systems and data-driven models for occupancy estimation, human participatory sensing, as well as the control algorithms for building actuation using real-time data from the wireless sensor networks and participatory sensing, and prediction for the data driven models. As part of this effort, we propose to analyze and evaluate the trade-offs between reducing space conditioning costs and providing quality of service to the building occupants.

1.1 Goals

1.1.1 Usage based Energy Savings

There are many opportunities to reduce energy usage through use-based management. The goal is to only utilize building systems when needed. For example, perhaps the most common use based management is controlling lighting through a passive infrared (PIR) sensor. Lights are actuated based on the motion detected by these sensors. Lights are only used when the room is occupied. This solution works arguably well for lights, with occasional issues when occupants sit motionlessly thinking about hard research problems. However, this crude binary sensing of occupancy is not always appropriate. Ventilation, for instance, is based on the actual number of people that occupy a room. In order to achieve efficiency, systems capable of monitoring occupancy in real-time with greater precision must be developed. Though the ability to adjust an HVAC system based on real time occupancy is an important step toward greater efficiency, perhaps just as important is the ability to anticipate room usage based on room usage. This needs to be addressed since conditioning a room is not instantaneous. For example, if it is known that a large number of people are in a lobby area, the HVAC of an adjacent conference room will be used with high probability, so it should begin conditioning the room beforehand. In order to achieve preemptive conditioning, we must be able to predict occupancy patterns.

1.1.2 Quality of Service

Recently many researchers have published works that explores how energy can be saved in buildings through better usage monitoring. However, few of these give occupant comfort a high priority. Often energy saving can come at the expense of the occupants. For example, during peak usage during the summer, facilities at our university have for brief periods increased temperatures at peak consumption hours during the summer in order to conserve energy. Energy is saved, but occupants may be overly warm. PIR sensors can turn off lights when rooms are not in use, but may sometimes leave occupants in the dark if they remain overly still. This trade-off philosophy, however, runs counter to the purpose of these systems; to make occupants comfortable. Taken to the extreme, if we want to maximize energy saving and do not care about occupants, then we could shut down all the lights and the conditioning system and achieve 100% savings.

Even when systems are not being optimized for energy savings, occupants are not always comfortable with baseline conditioning strategies. Determining occupant comfort levels is not straightforward with regard to what is a comfortable temperature. In order to address this issue, we argue that including the human in the control loop through participatory sensing provides a unique and effective method to improve quality of service to the building occupants.

1.2 Contributions

For both of these building management criteria, better sensing of the surrounding environment is required. Improving the conditioning efficiency of the building requires sensing of the occupants in order to condition rooms based on actual need. Improving the quality of service needs either additional sensing of environmental conditions or better communication with occupants. In both cases, wireless sensor networks can be utilized.

The goal of this work is to answer the following:

How can wireless sensor networks be utilized to achieve improved energy efficiency of buildings and quality of service for occupants?

In this dissertation, we contribute the following:

- Implementation of two different wireless sensor networks capable of monitoring occupancy in near real-time. One WSN utilizes low power cameras deployed in public hallways. The other uses thermal sensing in order to measure occupancy.
- Development of occupancy prediction models for use in predictive conditioning of spaces. For the development of these models, we also create large ground truth datasets.

- Development and testing of predictive conditioning strategies that incorporate real-time occupancy monitoring and occupancy prediction.
- Development of strategies that improve the quality of service and satisfaction of occupants using participatory sensing as well as models of thermal comfort and traditional sensing.

1.3 Motivation and Overview of Concepts

There are four major components discussed in this thesis that are used together for achieving greater efficiency and comfort.

1.3.1 Occupancy Monitoring

Currently, the majority of HVAC systems condition rooms assuming maximum occupancy during normal working hours and are turned off at night. This leads to inefficiencies as rooms are often conditioned to levels that are not appropriate for the number of occupants actually occupying the areas. An HVAC system could waste energy supplying ventilation enough for 30 people when only 10 actually occupy a room.

For demand response HVAC control, occupancy detection needs to be accurate, reliable, and able to capture occupancy changes in real time. Based on real time occupancy estimates, both temperature and ventilation can be adjusted based on actual usage.

1.3.2 Occupancy Prediction

Though the ability to adjust an HVAC system based on real time occupancy is an important step toward greater efficiency, perhaps just as important is the ability to anticipate room usage based on current room usage. This needs to be addressed since conditioning a room is not instantaneous and requires time for adjustments. For example, if it is known that a large number of people are in a lobby area, we want the HVAC system to know an adjacent conference room will be used with high probability and begin conditioning the room beforehand. To train these models, data from an occupancy monitoring system can be used.

1.3.3 HVAC Control

By combining an occupancy monitoring system with an occupancy prediction system, it is possible to develop more efficient control HVAC strategies. The occupancy monitoring system allows ventilation to be regulated in real-time based on the number of occupants within a space. Occupancy prediction allows a space to be pre-conditioned for temperature ahead of time when it is likely to be occupied.

1.3.4 Occupant Feedback

While energy efficiency is important, also critical is the comfort of occupants. The purpose of HVAC systems is to maintain comfort for occupants. Occupant feedback can be integrated with HVAC systems in real-time to better increase occupant comfort.

1.4 Organization

Chapter 2 starts with a summary of the relevant prior work that has been done with regards to the four major components. Figure 1.1 shows how these four components are related. We then follow with chapters developing each of the four major components.

We first start with the occupancy models component in Chapter 3. In this chapter, we develop four different occupancy models; a basic Gaussian model and three variations of a Markov Chain. We next develop two different occupancy monitoring systems in Chapter 4 and 5. Chapter 4 develops OPTNet, which is a wireless sensor network of cameras for estimating occupancy in near real-time. Chapter 5 describes our second generation wireless occupancy monitoring system, which uses thermal sensors along with passive infrared sensors to estimate occupancy. With methods of occupancy prediction and occupancy monitoring established, we next examine control strategies in Chapter 6. Chapter 7 and Chapter 8 utilize the systems, models, and control strategies we developed in the previous chapters. We show how these systems can be integrated with an actual building energy management system in order to actuate the HVAC system based on real-time and predicted occupancy estimates. In Chapter 9, we discuss quality of service issues. We introduce the participatory sensing network called Thermovote, which uses human as sensors in order to correct the error of thermal comfort models to create a more comfortable environment. We conclude with Chapter 10 where we summarize our conclusions and discuss the future research directions.

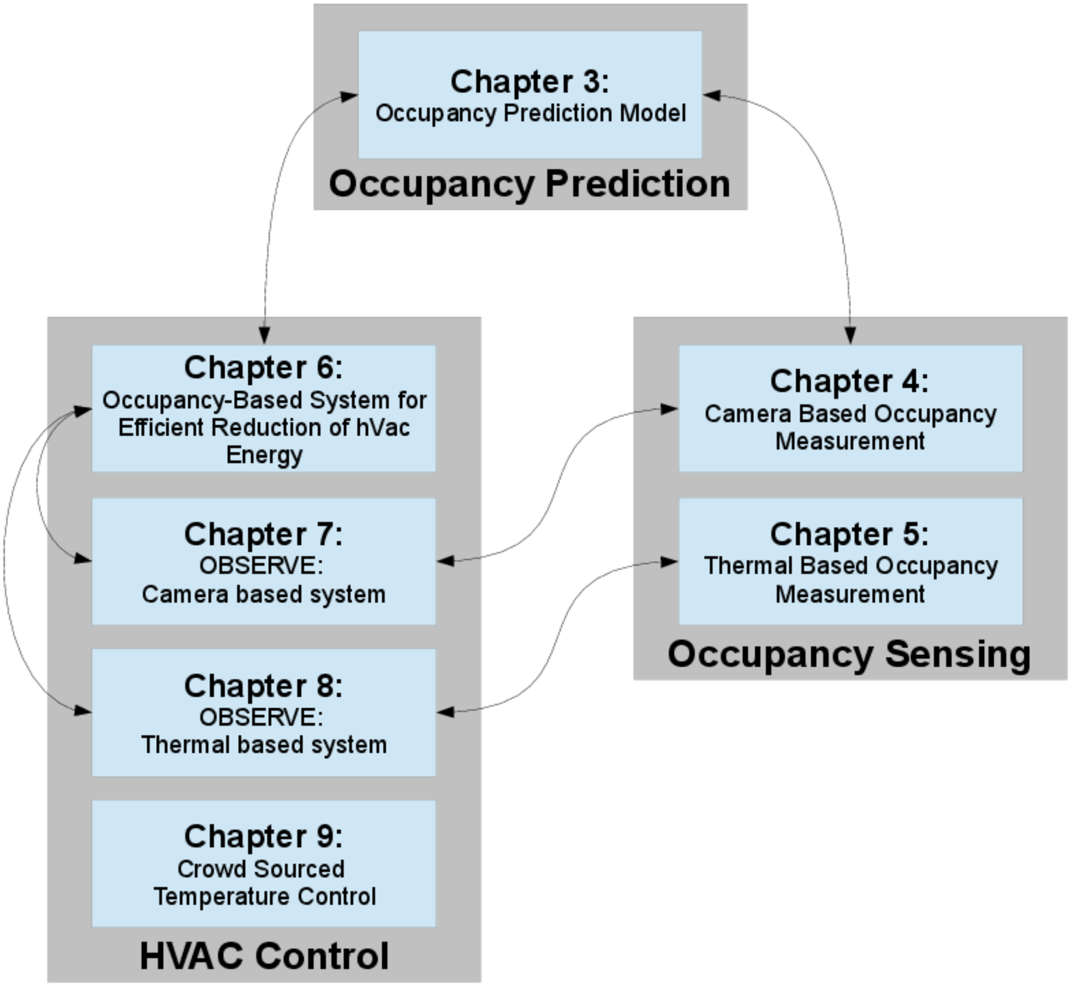

Chapter 3:
Occupancy Prediction Model
Occupancy Prediction
Chapter 6:
Occupancy-Based System for Efficient Reduction of hVac Energy
Chapter 7:
OBSERVE:
Camera based system
Chapter 8:
OBSERVE:
Thermal based system
Chapter 9:
Crowd Sourced Temperature Control
HVAC Control
Chapter 4:
Camera Based Occupancy Measurement
Chapter 5:
Thermal Based Occupancy Measurement
Occupancy Sensing

Figure 1.1: Concept Relationships

CHAPTER 2

Related Work

In the previous Chapter we discussed the major parts of a system that would be required to create a usage based system for HVAC control and management. We need a system for monitoring occupancy in near real-time, a method of predicting occupancy, an control strategy that combines occupancy monitoring and prediction, and finally a method of incorporating occupancy feedback ensure occupant comfort.

This Chapter is divided into two broad categories. The first section, Section 2.1, focuses on different methods of occupancy monitoring, occupancy modeling, and how occupancy monitoring and modeling can be incorporated into different usage based control strategies. The second section, Section 2.2, examines the quality of service of HVAC systems as well as other participatory sensing applications. We examine different works that measure thermal comfort, describe the various parameters relevant to occupant comfort, and implement strategies for maximizing occupant satisfaction.

2.1 Usage based Conditioning

In this section, we address the main problems related to developing a usage based heating, cooling, and ventilation system. At the most basic level, occupancy information is required; rooms need to be conditioned when occupants are present. There are two different indications of occupancy to be considered; binary occupancy and actual occupancy. A binary indication of occupancy simply indicates whether or not a room is occupied whereas actual occupancy indicates the precise number of people of a room. For systems such as lighting, a binary indication of occupancy is sufficient for control strategies. However, for ventilation, which is based on the number of people [ASH07b], the precise number of people is needed in order to optimally regulate ventilation. Section 2.1.1 focuses on the different methods of monitoring occupancy and discusses the implications of each method with respect to temperature actuation and ventilation. Section 2.1.1.1 discuss work related to image based occupancy monitoring and tracking for wireless camera networks.

Occupancy monitoring allows for reactive on/off actuation for systems such as lighting or ventilation. However for heating and cooling, conditioning reactively only when people are present is not suitable as a control strategy. The main problem with reactively actuating on/off temperature conditioning based on occupancy is that temperature response is often slow. When occupants first enter a room, they may be subjected to uncomfortable temperatures while the room comes to the temperature. To ensure room temperatures are comfortable when occupants enter a room, rooms need to be pre-conditioned before occupied. This can be done using occupancy prediction models. Section 2.1.2 examines different occupancy prediction models and how these models are integrated with predictive HVAC control strategies. For the remainder of this paper, reference to reactive control refers to on/off actuation of a system.

2.1.1 Occupancy Monitoring

Occupancy monitoring has been worked on for many years and long been recognized as a means of increasing energy efficiency. For over a decade, research has shown that occupancy based lighting control can save a significant amount of energy. The California Energy Commission showed that occupancy sensing actuating local lighting controls could save 25% to 50% for lighting energy usage [Com93] using passive infrared sensors (PIR) and ultrasonic sensors. These savings have been confirmed by other studies [GB00] [AD95]. However, there are also problems with this type of sensing. In a survey of 23 commercially available PIR and ultrasonic sensors, it was found that 18 did not register movement within the advertised specifications [NLP98]. Despite this, PIR sensors have long been a popular method of obtaining a binary indication of occupancy since they are cheap and accurate provided there is adequate coverage of the target area and if properly placed.

One technique for increasing the accuracy of PIR sensors is to supplement the system with different types of sensors. The authors of [LSS10] and [ABD11] utilize door and PIR sensors for binary detection of occupancy for residential buildings and examine reactive and predictive controls strategies. The door sensor is used to augment the PIR sensor in order to prevent false positives from overly still occupants. If the door sensor is activated, and the PIR sensor records movement shortly afterwards, then the state of the room is considered occupied. However, if there is no movement within a threshold of time after the activation of the door switch, then the room is considered empty. Both [LSS10] and [ABD11] showed that this door/PIR sensing to have superior performance than PIR alone for rooms with a single entrance. Prior to these works, this type of door/PIR sensing was also available as a wireless commercial product specific to HVAC control [Sol]. However, this method is limited to rooms with a single door.

Though binary indication of occupancy is suitable for applications such as lighting, heating, and cooling, having a more accurate estimate of occupancy allows for more opportunities to save energy. In particular, a more precise occupancy estimate allows for the control of the ventilation, which is adjusted based on the number of occupants [ASH07b] or CO_2. Reducing the amount of ventilation helps to save energy by reducing the amount of outside air that needs to be conditioned [EPS01]. As illustration, it is more efficient to condition a room with all the windows closed than to condition a room with the windows open. The same principle applies to HVAC systems except a minimum amount of ventilation is required based on the number of occupants. In [EPS01], the authors report that 6.8% to 20% energy savings are possible by optimally controlling the ventilation. In the following sections, we examine several methods of occupancy estimation and will consider ventilation as the target application. In addition, we will also take into account deployment and maintenance issues.

One strategy used is to attempt to estimate occupancy implicitly from existing infrastructure. The papers [MRN11, TYS13] uses the existing network in order to estimate occupancy. In particular, the authors discuss many different methods of using existing infrastructure that could potentially be used for occupancy estimation, such as phone call volume, mouse and keyboard detection (sent over network), and security door access records. However, for their actual experimentation, they simply use Address Resolution Protocol (ARP) to count the number of occupants. They first use ARP to classify each host connection as Ethernet or WiFi. For WiFi connections, they use triangulation in order to determine the locality of occupants. Thus, for each area, the occupancy estimate is the total number of hard-line connections and the estimated WiFi connections. Though many of the methods they discuss are valid for the purpose of occupancy estimation, many have too many privacy issues, tendencies to undercount, and deployment complications for a practical solution; these issues are not addressed in the paper. The fact they only use ARP suggests difficulty obtaining permission to use such sensitive data. If we only consider ARP, then the main drawback to this method is the assumption that occupancy can be determined solely based on network connections. Occupants could have a connection but be away from their computer. Not every occupant may have a connected machine and some may have multiple computers connected. In order for implicit estimation of occupancy to be accurate, the measured phenomenon must be universal and consistent among occupants.

The paper [BXN13] takes a similar approach of utilizing WiFi connections from cell phones for estimating occupancy. The authors assume a zone of detection radius for each access point (AP) and associate each AP associated with one or more area. Since localization was not accurate enough do make the distinction between public and private spaces, they always assume public spaces are occupied and focus on using WiFi for detecting occupancy for private spaces. For each private space, they associate an AP and an occupant connection uniquely identified using the MAC address. In order to achieve these mappings, they require users to supply this information. They assume that if an occupant is connected with their associated AP, then they occupy their private space. There are multiple drawbacks to this approach. They require users to maintain continuous connections to the AP and must set their phones to fetch mail every 15 minutes to maintain the connection. Another major drawback is that this approach does not address shared spaces. Since shared spaces are always considered occupied many locations, such as conference rooms, that are often unoccupied are considered always occupied. Maintaining the space, AP, and occupant mapping is also an issue; occupants can change phones and forget to update their mapping information. Similarly, the same issues arise if the AP gets replaced or if an occupant no longer "owns" the space.

The authors of [AC01] show how occupancy can be estimated from lighting and equipment loads. Data was collected for lighting and equipment loads and occupancy is estimated using a walk-through survey of the building. Using the electrical loads as the indicator variables and occupancy estimations as the response variable, linear regression models for weekdays, weekends, and holidays were created. The main limitation of this model is the reliance on energy usage to determine if someone is present. In general, this will tend to underestimate occupancy levels, which could potentially lead to under-ventilation. It is important to not that while over-ventilation is inefficient, it is safe. Under-ventilation, however, is not. For example, occupants attending a large group meeting in a conference room may not be adding any additional loads, causing the model to report the room empty. Similarly, there are instances when appliances are left on, skewing the result of the model. From a deployment perspective this strategy is also problematic since it requires monitoring at the plug level. This requires additional infrastructure at the plug level or breaker level. Another deployment issue is making sure that the location of the equipment does not change.

The paper [KBS13] also used a load based approach for determining a binary indication of occupancy. In this paper, the authors examined load profiles for 5 houses over a period of 8 months. Ground truth occupancy was gathered via a tablet placed at the entrance of each house where occupants can mark their presence. In addition, PIR sensors were also deployed for an addition method of verification. In order to establish correlation between occupancy and the load, multiple "smart" plugs capable of measuring individual device loads were installed along with a meter for the total house load. The authors test multiple classification techniques in order to determine if the house is occupied. The best classifications methods, Support Vector Machine (SVM), K Nearest Neighbors (KNN), and Hidden Markov Model (HMM), produced accuracies between 63%-93%. The main drawback to this method is the training requirement. In order for these classification techniques to work, ground truth data must be gathered for the target building. Given that PIR sensors can provide accuracies greater than 90% in most cases if properly deployed, one is better off using a PIR sensor rather than load data for a binary detection of occupancy. In [CBS], the authors developed a method of measuring occupancy in residential homes using load data alone without additional sensors such as PIR. In this paper, they used a threshold based approach simply using the average and standard deviation of load data in order to calculate a threshold for occupancy. This strategy yielded an overall accuracy of 73%-90%.

A common strategy is to incorporate multiple sources of sensor input in order to infer the occupancy of a space. The authors of [DHT06] place three PIR sensors in each office and uses on/off hook sensors on the telephones. They then use a belief network of the multiple sensory inputs to determine the occupancy of the room. There are several drawbacks to this approach. Though multiple telephone sensors could potentially be used to help determine a lower bound of occupancy, this system does not account for people occupying the room who do not have a phone. For example, a private office only has one phone, but it is likely that meeting in a private office will have one or more additional occupants. With this system, is also difficult to determine when someone leaves an area occupied by several people since the presence of multiple people essentially nullifies the usefulness of the PIR sensor. For the purpose of ventilation, this would require providing additional ventilation to provide larger margin of error in order to minimize the risk of under-ventilation.

Rather than relaying on the more traditional PIR sensor, the authors of [SBK11] instead utilize active radio frequency identification (RFID) tags to determine occupancy. Tags were attached to the keys of the occupants of 5 homes. When within the range of the antennas, the RFID tags transmit every 5 seconds. Though this worked well in the majority of the cases, there were instances when errors occurred when occupants did not leave the house with their keys. This method also requires all occupants of the house to possess an RFID tag when leaving the house. If a tag is left behind, then the house will erroneously detect an occupied state. Similarly this method does not take into account guests who do not have RFID tags; if ventilation is considered, then every occupant must have an RFID tag to accurately estimate occupancy.

[LHD09] estimates occupancy by measuring a variety of parameters. In their deployment, the authors measure CO_2, CO, lighting, temperature, humidity, motion, and acoustics. For each parameter, they define multiple feature vectors: the original output, first order difference, second order difference, and a 20 minute moving average. These feature vectors are then used with three different approaches to estimate occupancy; a support vector machine (SVM), an Artificial Neural Network (ANN), and a Hidden Markov Model (HMM). The authors of [MMC11], also used a similar approach. For their deployment, the authors use CO_2, PIR, humidity, and acoustic sensors. Rather than simply using only differences for the feature vectors as was done in [LHD09], they also consider the average baselines over differing lengths of time as well as the time of day the readings were observed as part of their set of feature vectors. They test various combinations of their feature vectors with linear regression, logistic regression, perceptron, and SVM's to learn the occupancy levels. The paper [EBV13] uses CO_2, room temperature, and ventilation actuation signals to estimate occupancy. The authors test several approaches to estimate occupancy using these parameters. The authors first approach the problem by identifying a Linear Time Invariant (LTI) dynamic model using ground truth data that takes ventilation and occupancy as inputs in order to predict CO_2 concentration and room temperature. Once the model is identified, then it is possible to solve for occupancy given CO_2 levels, room temperature, and ventilation rates. They also tested the use of a Support Vector Machine and Neural Network to estimate occupancy using the other parameters as inputs.

Though the results from [EBV13, MMC11, LHD09] seem to show these methods work reasonably well, there are several issues that make the results suspect. There is a fundamental problem using CO_2 sensors for occupancy estimation for the purpose of ventilation that these papers do not take into consideration. They assume that ventilation will not affect occupancy estimates of the room. However, since ventilation will affect CO_2 levels and thus occupancy estimates, we argue that using CO_2 sensors in combination with ventilation will lead to wild fluctuations in ventilation actuation and periods of under-ventilation. CO_2 sensors are notorious for having a slow response time and their need for careful calibration [FFS06]. Suppose other inputs along with CO_2 levels of a room indicates that the room is occupied by multiple people, requiring that ventilation occur. As ventilation occurs however, the CO_2 levels drop causing the occupancy estimate to also drop. Since there is an occupancy drop, ventilation decreases. This in turn causes CO_2 levels to increase and again increases occupancy estimates. This process is repeated as long as occupants are present. During the periods where the occupancy artificially appears to be low, the room will be under-ventilated until the CO_2 sensor slowly detects the CO_2 buildup. It may be that one would be better off using CO_2 sensors to regulate ventilation directly since it would only under-ventilate once when rooms are first occupied. In addition to these reservations, there are several aspects that make these solutions less attractive from a deployment perspective. While increasing the number of sensors also increases redundancy, it also increases the initial cost of the deployment as well as on-going maintenance cost; more sensors need to be checked and calibrated for the lifetime of the deployment.

2.1.1.1 Wireless Camera Occupancy Monitoring

Since our work uses camera based occupancy monitoring, we will examine several efforts that use wireless camera networks and will also explore relevant image processing techniques. Commercial optical counting systems exist [tru] for wired deployment. In addition, others have successfully used wired camera system that have a high degree of accuracy with regards to counting or tracking occupants [BLE11]. However, these systems are more costly and difficult to retrofit for buildings and are designed to require hard wiring. For our system, we wish to utilize motes that can be easily deployed in even older buildings. With efficient use, mote battery replacement can be done along with routine light-bulb replacement.

Over the past few years, several low-powered wireless cameras have been developed [CAB08, imb, KJD09]. One major limitation of wireless camera networks is that the computational power and memory of wireless embedded systems are limited, which limits the techniques that can be utilized. In this section, we examine several wireless camera networks developed for occupant tracking and counting.

In [KJD09], the authors developed a 16 sensor node deployment within the hallways of a building. Each node at transition boundaries consisted of 3 cameras. Since individual nodes lacked the memory and processing power to operate in real-time, collections of three Tmote/Cyclops cameras were used at each boundary where each mote took turns capturing the sensed area in a round robin fashion. This round robin strategy was used overcome the memory and computational constraints of the motes. While one mote captures images, another mote processes images it just captured. The collection of cameras capture and process the image data at each boundary to determine if a transition occurred. While the system was shown to be accurate for counting individual components, additional tests over 24 hours showed that the system is only able to capture $\approx 80\%$ [ECC11]. In particular, this paper does not address how errors in transitions lead to cumulative errors in occupancy.

The authors of [TS07] use Imote2 motes [imb] with Enalab cameras to localize and count occupants in areas. For their deployment, cameras are mounted overhead with fish-eye lenses to maximize the view area and avoid occlusions. They detect foreground pixels by performing background subtraction using the previous frame as a background. This makes foreground detection sensitive to movement and essentially assumes that occupants are always in motion at least at some point in time. In order to localize each occupant, they develop the concept of a motion histogram. The frame is subdivided into a grid where each grid position represents a block of pixels. A fixed bin, larger than the grid size, is defined based on the size of the of a person and the defined grid pattern. A sliding window algorithm is then performed with the bin over the grid positions. At each position, the number of foreground pixels the bin covers is counted. A running total of foreground pixels is kept at each grid position. Since the bin can cover the same grid positions multiple times as the bin is slid across the image, the same foreground pixels can be added to the associated running total. This process constructs a two dimensional histogram where the peaks represent the most likely position of occupants. In [LBS09], the authors monitor elderly people using Imote2 motes with Enalab cameras. To track occupants, they also use a motion histogram. Occupant counts are achieved by counting peaks within the histograms. These peaks are also used to estimate the motion of occupants. Their system also includes PIR sensors in order to detect occupancy for certain areas. There are several drawbacks to these approaches. The system assumes that the camera can actively monitor the entire area of interest. The problem of privacy also exists for these systems; cameras must be placed directly in the room.

2.1.2 Occupancy Prediction

While binary methods of measuring occupancy have been utilized for actuating lighting for many years, little work has been done to actuate HVAC systems using occupancy data. Some hard wired commercial products are integrated with the building controls for a similar reactive control and the latest version of California Title 24 energy code [Tit13] now requires demand response ventilation. The authors of [ABD11] condition spaces by reactively adjusting the temperature based on a binary indication of occupancy. In order to maintain a minimum amount of conditioning, they establish upper and lower set-points according to the standards set by [ASH07a]. However, there are several critical drawbacks to this reactive approach. The main problems with actuating HVAC based on real-time data alone is that conditioning is not instantaneous. Time is required for a room to come to temperature. If the temperature of an unoccupied room is allowed to “float”, then the temperature of the room could be potentially quite uncomfortable when the occupants initially enter the room. If the room is large, then the duration discomfort could be protracted. Since building energy managers wish to minimize complaints and maximize comfort, most managers would be reluctant to subject occupants to discomfort in order to save energy as well as violating standards of thermal comfort [ASH07a]. In the case of [ABD11], the location was in San Diego where the mild weather naturally reduces thermal discomfort since outside temperatures are more likely to match the relaxed setback temperatures.

2.1.3 Static Occupancy Models

The simplest predictive HVAC control strategies are applications where the occupancy schedule is static and known beforehand. Examples of this scenario are schools where the number of student occupying each classroom is known depending on the time of the day or hotels where the number of guests in each room are known based on the registration. Since occupancy is not actually monitored through sensory input such as PIR, energy efficiency comes from optimal time-of-day HVAC scheduling. By optimally scheduling the HVAC system, the authors of [MAA01] demonstrated that 25% - 46% energy savings were possible through HVAC scheduling depending on the usage type of the building. Buildings with less predictable occupancy schedules are more difficult to opportunistically schedule the HVAC system.

In situations where occupancy patterns are unknown, it is common to use static occupancy models derived from other similar buildings. These models are sets of predefined static coefficients that are multiplied with a maximum room occupancy. To estimate the occupancy of a room at 8am given that the room has a maximum occupancy of 30, the coefficient for 8am is simply multiplied with

30. Different sets of coefficients are used depending if it is a weekday, weekend, or holiday and the purpose of the building. ASHRAE Standard 90.1 [ASH07c], BLAST [HL83], and DOE-2 [Doe12] define several occupancy profiles for office buildings day-types. These models are commonly used for energy simulation tools such as eQuest [Bui] or EnergyPlus [Epl] during the design phase of buildings and are also used to determine static conditioning strategies for buildings. These models do not accurately capture changing trends in usage and generalize poorly.

Capturing the schedule of a building can be solved with a learned occupancy prediction model using data specific to the target building. By utilizing data from the actual building, better scheduling is possible and changing dynamics can be captured. With a model, one can probabilistically determine if and when a room will be occupied and the room can be pre-conditioned accordingly. Initially, few works on demand control utilized occupancy modeling were published. Most still acknowledge that a major obstacle for developing occupancy prediction models is difficulty gathering ground truth data. It is time-consuming to gather ground truth occupancy data even for a small set of rooms. With new systems of monitoring occupancy however, it is now simpler to develop occupancy models and capture changing trends. In the following sections we will focus on common methods of prediction suitable for occupancy and predictive conditioning.

2.1.4 Time Series Forecasting

Time series has long been used in areas such as econometrics and signal processing. The dependency of occupancy change on time in buildings suggests it may be possible to model occupancy using time series techniques. Intuitively, occupancy will change based on time dependent events, such as arrival in the morning, leaving for lunch, delivering mail in the afternoon, and so forth. In this section, we discuss commonly used time series techniques that could be potentially used for occupancy forecasting.

Consider occupancy of offices that operate year-round. If we examine the data at weekly, monthly or annual range, then there is no trend. However, there is a great deal of seasonality related to the hour of the day; people arrive in the morning, spend their day working in a space, and then eventually leave.

The most basic technique is the autoregressive (AR) model [BJ90]. Let x_t represent the observed occupancy at time t. For this technique, it is assumed that the x_t is based on the previously observed occupancies x_i where $i = 1 \ldots n$ and n is the number of observations into the past to consider. The length can be adjusted to capture periodic behavior. The value of x_t is then defined as

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + \epsilon_{wp} \tag{2.1}$$

where α_i is the weight for x_{t-i} and ϵ_{wp} is white noise. For this method, it is assumed the future occupancy can be estimated from the previous p steps of occupancy where each step is assigned a weight. This model is defined as $AR(p)$. Another basic standard technique is a moving average (MA) model:

$$x_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} \tag{2.2}$$

where $\epsilon_t \ldots \epsilon_{t-q}$ are white noise error terms, $\theta_1 \ldots \theta_q$ are model parameters, and μ is the mean of the series. This model, denoted as $MA(q)$, assumes the observed value is a random noise term plus a linear combination of the previous q noise terms.

The ARMA model [BJ90] combines the AR and MA models previously described:

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_n x_{t-p} + \epsilon_{wp} + \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_n \epsilon_{t-q}. \tag{2.3}$$

Since this model incorporates both AR and MA, it also inherits the same problems. A requirement of AR and MA and is that the data is stationary; the same holds true for ARMA. After testing these methods, we found the strong seasonality of the daily occupancy pattern made these approaches not ideal for our application.

We next tried the more general model ARIMA [BJ90] (also referred to as Box-Jenkins), which is an extension of ARMA where differencing is used to de-trend the data to transform a non-stationary time-series into a stationary distribution. Typically it is referenced as the function $ARIMA(p, d, q)$, where p, d, and q are the order of the AR ($AR(p)$), integrated ($I(d)$), and MA ($MA(q)$) parts respectively. The $I(d)$ differences the observation by d lags, which depends on the linear trend observed. Since daily office occupancy does not have a linear trend over multiple days, we found the ARIMA model modeled on this time scale not to be appropriate; the differencing would not remove seasonality. One potential strategy to use ARIMA would be to examine specific trends within each day. Examining occupancy at the daily level, there is no linear upward/downward trend since buildings eventually empty at the end of the day. At the hourly level, however, we can observe trends such as the increase in occupancy in the morning and the downward trend at noon when people go to lunch. For each of these trends, we can also observe "shocks" to the trend from events such as people going on breaks or moving between spaces. We attempted to subdivide the day into multiple parts and model these individual trends. While this piecewise strategy is somewhat feasible, it requires knowledge of upward/downward events and assumes these events do not change day to day. This is difficult to capture and we found that this assumption often did not hold true. Identifying events and generalizing events is a non-trivial. We also found it leads to over-fitting if the modeled intervals are too small. In addition, subdividing the day did not help us determine when occupants would occupy rooms without distinguishable occupancy patterns.

As mentioned previously, we found seasonality prevented our use of the previous models; seasonality causes a non-stationary mean that cannot be removed with differencing. Since occupancy data has a strong and stable seasonal component, seasonal ARIMA is a better candidate than the previously discussed approaches. This method includes additional terms that captures seasonality. In addition to having the terms $ARIMA(p, d, q)$, it also includes the terms for the model $ARIMA(P, D, Q)$, which are multiplied with the terms of $ARIMA(p, d, q)$. Under this model, the $AR(p)$ and $MA(q)$ are used to model the short term correlations; the $AR(P)$ and $M(Q)$ are used to capture the seasonality. This model is written as $ARIMA(p, d, q) \times ARIMA(P, D, Q)_s$, where s denotes the seasonal lag (24 hours in our case).

Another strategy that accounts for strong periodicity is describing the data through a linear combination of sines and cosines [Ham94]:

$$x_t = \sum_{k=1}^{q} [U_{k1} cos(2\pi\omega_k t) + U_{k2} sin(2\pi\omega_k t)], \tag{2.4}$$

where U_{k1} and U_{k2} are independent zero mean random variables with variance σ_k and ω_k are frequencies ($k = 1 \ldots q$). Because of the high degree of periodicity in the occupancy data, different combinations of the sines and cosines could be used to capture the dynamics found during different parts of the day. An advantage of this method over the seasonal ARIMA is that less parameters are required. However, as with the other tested methods, we found this method did not work well for areas that are not consistently occupied.

Though there are no references that explicitly describe the use of these methods for occupancy modeling in buildings, we discuss several works from other areas that examine the hourly dynamics with seasonal components. The authors of [HLJ09] test several time-series methods to forecast patient volume in order to manage hospital space, resources, and personnel. In their paper, they test two different models: a seasonal ARIMA using a 24 hour lag to capture the seasonal component, and a sinusoidal model. These methods were then used for 4 and 12 hour forecasts at three different locations. Both models performed well and outperformed predictions based on historical average. When comparing the seasonal ARIMA and the sinusoidal models, they found both to forecast equally well for their application. This work has many similarities to occupancy since the seasonality also operates on a 24 hour period. The paper [WH03] uses seasonal ARIMA to forecast car traffic flow. In their work, they compare the seasonal ARIMA with historical average and a random walk forecast. They define a random walk forecast as follows: the next predicted values is the last observed value. Seasonal ARIMA was found to have the best overall forecast performance. While applicable to this particular application, hospitals have specific patterns that are different than those observed in offices, labs, or even houses. For example, patients tend not to come and go daily nor they go out for lunch.

2.1.5 Markov Models

Another strategy for occupancy prediction is to frame the problem as a Markov process. The general strategy is to assume the state of the room occupancy depends only on the previous state of occupancy. Several papers in the literature explore this strategy. The paper [DHT06] defines a Markov Chain model for the binary occupancy of individual rooms. The states of the model are occupied or unoccupied and the transition matrix probabilities are calculated by examining the exponential distribution of the sojourn times of the observed states. The sojourn times are found using the firings of the PIR sensors. One drawback to this approach is that the model assumes the occupancy states of rooms are independent. However, in reality, the occupancy of adjacent rooms have at least strong correlations. Also not taken into account is the time of day. The occupancy state of the room depends on the time of day, which would change the transition probabilities. For example, the probability of an occupant exiting the room is higher at noon since he/she is likely going to lunch. This method also does not take into account differences between weekdays and weekends or different seasons.

The authors of [WFR05] uses a similar approach but also take into account the time of day. They start by modeling the occupant state of single offices using a non-homogeneous Poisson process using two exponential distributions; one for the occupied state and one for the unoccupied state. They first attempted to subdivide the day into hourly intervals and then determine the occupied/unoccupied distributions for each hourly interval. However, they found many of the fits to be not significant. Based on these results however, they subdivide the days into three events; morning arrival, leaving for lunch, and leaving work. The occurrence of each event is estimated using a normal distribution where the time of the event occurs within 1 standard deviation. In between each event, they use occupied/unoccupied exponential distributions to model the occupancy dynamics, which they found to be a Poisson process. While this technique does incorporate time, hand crafting the states using events can lead to over-generalization; it assumes that these events always occur. This model would not be applicable to other buildings such as residences where occupants are generally away for most of the day. One would need to handcraft a specific model based on domain knowledge for each type of building. Even then, there are many instances where events may not occur. A part time employee may only arrive in the afternoon. A conference room may only be used one day a week for meetings. Lastly, this model assumes the adjacent room occupancies are independent and does not attempt to gain additional information through the use of these rooms.

In [LSS10], the authors use a Hidden Markov Model (HMM) to achieve occupancy prediction. The hidden variable is the distribution of the home being in one of three states: away, active, and sleep. The HMM transitions to a new state every 5 minutes. The observed variables are the time of day (4 hour resolution), the number of sensor readings over a time interval, the binary readings of the doors switches, and the binary readings of the PIR sensors. Again, the drawback to this approach is the handcrafting of the states. While this may be applicable to residential buildings, a different set of states would be required for a commercial building, which has more a complex variety of possible states. Another problem with this approach is that it does not account for different daily schedules. Since the distribution is based on all of the observed days, no distinction is made between different days of the week. This will cause issues if events only occur on specific days. For example, a family may consistently return home at 3 pm every day except Wednesdays where they go to soccer practice and return at 7pm. In this situation, the system will likely still assume a 3pm return on Wednesdays and condition the house needlessly. Similarly, if a person stays out late on Friday on a semi-regular basis, the model would not be able to predictively condition for this scenario.

The authors of [DA09] attempt to avoid the handcrafting of states by using Episode Discovery [HYC04] to learn the common states and then use these states with a Semi-Markov Chain. The Episode Discovery algorithm examines the sequence of sensor events over a sliding window and then determines what patterns are significant from several days of sensor data. Each significant pattern is defined as an event state that is used with a Semi-Markov Model. Similar to the other MC approaches, an exponential distribution is used to model the time spent in a state before jumping to one of the other learned event states. For their model, they consider adjacent room occupancies to be independent. What is not clear from the paper is what happens when a sensed state does not match one of the other known event states. In this scenario, one would encounter a discontinuity.

The method closest to our work is proposed by [RTI08]. In their work, they attempt to model occupancy for residential homes. For a target building, they assume a maximum occupancy n and create an $n \times n$ transition matrix where $n = 1 \dots 7$. In order to account for time, they create a transition matrix for different windows of time. In total, they construct 144 transition matrices (10 minute windows). While their strategy is suitable for low occupancy single zoned buildings, their approach does not scale for large multi-zone buildings. A non-trivial time is required to learn these transition matrices. The framework does not account for inter-zone correlations and assumes that state continuity exists between the time slots; from our experiences described in Chapter 3, this is not the case for areas with large populations. Our research in Chapter 3 discusses these issues in detail and proposes methods of solving these problems.

2.1.6 Classification Based Prediction

Through proper framing of the problem, it is possible to create prediction models using a classification framework. The authors of [SBK11] developed a prediction based conditioning strategy for residential buildings using k-nearest-neighbors (KNN) to predict occupancy. The binary occupancy schedule of each day is recorded and stored for later comparison. To predict the occupancy, the occupancy schedule up to the current time is compared with the same time-frame from all previously observed days. The 5 closest days, determined using Hamming distance over the occupancy vector, are then used to calculate the probability that a space will be occupied a specified amount of time into the future. In order to increase the accuracy of the prediction, a distinction is made between weekdays and weekends when computing the distances. This prediction method worked reasonably well overall with accuracies ranging from 80% to 86% for a 90 minute prediction. However, the method did not perform well for rooms that were not consistently occupied and tended to predict these rooms to be empty. Despite this, the MissTime [LSS10], which is the total amount of time occupants are subjected to a temperature 1°C outside the specified heating and cooling set-points during normal waking hours, showed a 38% to 88% improvement over a static schedule strategy.

The use of KNN creates several drawbacks. KNN works best if the number of days in the dataset is small since KNN is simply a comparison among all points in the dataset. If the dataset of days is large, then the time required to compute the prediction may be prohibitively long. While it may be possible to sub-sample, it is possible that even the number of sub-samples required for a reasonable accuracy could still be too large. One possibility is to use other classification methods such as support vector machines (SVM) or artificial neural networks (ANN). For these methods, however, some adjustment of the framework is necessary. The KNN method is able to use an input vector of occupancy of any length; when the comparisons are done, the input vector is checked against the corresponding time frame of the other days. However, SVM's and ANN's need to be trained to accept a fixed number of inputs and provide a fixed number of outputs. This could be done by training multiple models over different periods of time and provide a prediction a fixed amount of time into the future. For example, a model could be trained to take occupancy of the past hour as an input and provide a prediction an hour into the future. Training multiple models would take time initially, but this training would be only be required once. A similar strategy was utilized by the authors of [DA09] to model occupancy for individual rooms. In order to predict occupancy for a particular time t, they train a model using the room data from midnight up to t from previously observed data. This strategy was applied using Multilayer perceptron and logistic regression using data that was gathered through user surveys. These methods were able to predict occupancy with an accuracy of 67%-75%. Similar to the other papers, they model room occupancy independently and do not consider inter-room correlations.

2.1.7 Predictive Conditioning

Though many different models of occupancy prediction exist, few works actuate heating and cooling based on occupancy prediction models and those that do are often done with static prediction. This is in part due to the lack of occupancy sensing infrastructure and a means of automated actuation. With the introduction of wireless sensor networks and an increasing number of commercial automation retrofitting products, more work is being examined in this area.

Thus far, almost all the work that has been implemented has attempted to approximately optimize the energy consumption at the local level. However, optimizing energy at the local level may not guarantee optimization of energy at the global level. Techniques such as Model Predictive Control [CB04] or Markov Decision Process [Put94] could potentially be used to optimize energy globally. Since these techniques would require a detailed thermal model describing the heat transfer interactions among all the rooms as well as from other outside factors, these global optimization techniques are not within the scope of this paper.

In [LSS10], which uses Hidden Markov Model (HMM) for occupancy prediction, the authors provide results for real world deployments and EnergyPlus simulations. The authors of [SBK11], using their KNN prediction approach, tested their predictive conditioning strategy on 5 houses with an average of 61 days per house. In order to test both central and de-centralized heating, three of the homes were in the US (central system) and two in the UK (de-centralized individual room heating). Since ventilation was not considered, they utilized a binary prediction of occupancy. Interestingly, the strategy did not work well with the centralized systems when comparing the predictive strategy to a typical static schedule strategy. In two of the homes, there was actually a decrease in gas efficiency. However, the UK homes did save 8% and 18% in gas usage when scheduling on a room by room basis.

[GK] employs a similar technique utilizing KNN for occupant prediction. In addition to determining a binary indication of occupancy, they also measure the radiant heat of occupants in order to better measure occupant comfort. Based on the occupant comfort, predicted occupancy, and thermal properties of the room, they solve an optimization problem minimizing occupant discomfort while maximizing energy savings. In addition to the same drawbacks of [SBK11], [GK] does not address the prediction accuracy of occupant comfort, which is critical for their control scheme.

[DA09] uses their Semi-Markov model from their learned states to test predictive condition strategies in an EnergyPlus simulation. They test four different methods of conditioning; a fixed 7am-6pm schedule, an email client meeting scheduler, reactive motion actuation, and a the Semi-Markov predictive approach. For their predictive approach, they first examine the current state of the system. If it is likely that a room will be occupied a significant amount of time in the future, then the room is conditioned. Using this predictive method, they were able to save 30% in energy costs. There are several weaknesses to their approach. Again the use of CO_2 is questionable. Another issue is that their framework does not take into account inter-room correlations.

2.2 Quality of Service

The most common method of determining thermal comfort is by trial and error. Building managers will often guess a temperature and then take into consideration complaints from the users. However, there are more precise methods of determining comfortable temperatures. Multiple works have explored different models of thermal comfort[Hop99, ASP97, Gag86, AZHa, AZHb, ZHA, HZA]. Within the building community, the most commonly used metric for thermal comfort is [O70]. This paper defines the Predicted Mean Vote metric (PMV), which establishes a model of thermal comfort using humidity, temperature, radiant temperature, metabolic rate, clothing insulation, and air flow as the parameters. Based on these factors, the model returns a value between -3 (cold) and $+3$ (hot) where 0 indications ideal thermal comfort. Since then, many other papers have examined how temperature impacts thermal comfort [SBW10, PHK07, PHK10].

2.2.1 PMV Measurement and Actuation

Several systems have been implemented to measure PMV. Authors of [YYC03] developed a specialized comfort sensor that measures air temperature, air motion, mean radiant temperature, and humidity. Input from these sensors are then used with PMV or standard effective temperature (SET*) models. They examine how their sensor choice affects the estimates of PMV as compared with ideal sensors. The main drawback to their work is that all their results are done in simulation; there is no verification that compares the sensed comfort level with the actual comfort level. Though their sensor may yield more accurate estimates for the PMV and SET* models, it is unclear if this will also reflect actual comfort levels. For factors such as air speed, it is unclear how many of these sensors are required for adequate estimates of thermal comfort.

The paper [TC07] describes another system of measuring thermal comfort in real-time. They separate their system into separate sensor nodes; temperature sensors, radiation sensors, and PMV sensors. Each node in the system is designed to accomplish different tasks locally. The temperature sensors measure the temperatures of the surrounding surfaces. The radiation sensors measures the solar radiation absorbed by occupants. The PMV sensor measures air temperature, air velocity, and humidity. In order to reduce the computational load of the PMV sensors, a table of PMV values are precomputed with different combinations of the parameters. For the metabolic rate and clothing insulation are given initial values, but can be modified manually by occupants. Since occupants are allowed to adjust insulation and metabolism in order to adjust for PMV estimation errors, it seems that asking users directly for PMV value is a less error prone since occupants are more easily able to describe their thermal comfort level than metabolism or insulation.

In the paper [RA09], the authors propose an architecture to measure thermal comfort. They utilize SunSPOT nodes to measure air temperature. They assume mean radiant temperature is equal to dry bulb temperature and for the remaining parameters, they assume fixed values. Sensors are placed around the room in a distributed fashion and used to estimate the PMV value at each location. They use these values in order to estimate the PMV value at unknown locations within the room. This is done by weighting measured values by the inverse of distance to the unknown location. For this approach, sensor coverage is again an issue to consider. They do not address how much coverage is required or how sensor coverage could potentially affect the error derived from the interpolated values. Another drawback to their work is that they do not compare their results with actual occupant comfort.

Another method of improving PMV estimations is to measure other environmental parameters rather than assuming fixed values. The authors of [GK] accomplish this by creating a turret system that can track individual users using a Microsoft Kinect [kin] coupled with a thermal sensor. With this tracking system, they are able to estimate the clothing level and increase the accuracy of the PMV calculated using Fanger's model. There are several drawbacks to this approach. Their system can only track one person at a time, making it only practical for single occupancy areas. For areas with multiple occupants, the number of turrets makes this strategy impractical. Another issue not addressed in the paper is how much this single parameter impacts the overall accuracy of the PMV. Given the complexity of the system, it may not be worth the effort of deploying such a system in order if the accuracy gain is minimal. On a qualitative level, occupants may not feel comfortable with a Kinect tracking their movement while in a room.

The paper [HW], developed after our work was published, has many similarities to our work. They have a similar cell phone application where users input their thermal comfort on a 7 point scale and use this input to change temperature set-points. Rather than use the Fanger PMV formula for determining temperature, they instead attempt to learn individual user comfort levels using the room temperature, outside temperature, and user votes. Once each model is developed for each person, they calculate the optimal temperature by minimizing thermal discomfort through each user's individual preference. There are several issues that are not addressed by this paper. Based on our work, we found that outside temperature has little correlation to indoor PMV and the authors do not address the error for their models for each individual. Another issue we have encountered that is not addressed is the bias that is introduced due to internal competition; occupants often exaggerate their discomfort in order to offset the votes of other occupants. This implies the individual models will often be flawed.

2.2.2 Participatory Sensing

A major component of wireless sensor networks is the acquisition of data. Typically this is done using a traditional deployment of standard sensors. Participatory sensing [SGS] is a method of data collection that leverages off the prevalence of mobile devices. Since many mobile devices now have a wide variety of sensors available such as GPS, cameras, accelerometers, microphones, as well as others and are already widely distributed, there are those who are exploring how to exploit this ready made sensor network deployments for data acquisition. In addition to gathering data from mobile sensors, these networks have the unique advantage of having a live person as a potential sensor. Many sensing problems normally impossible using traditional sensing can be done by using humans as sensors.

Several participatory sensing applications have been developed in recent years. PEIR (Personal Environmental Impact Report) [PEI, SGS] attempts to classify a person's activity based on locality found using GPS and cell phone tower triangulation. Based on the activity classification, a persons environmental impact can be estimated. Biketastic [RSD10] uses mobile phones to collect information about bike routes used by bicyclist in the Los Angeles area. For this application, information such as user contributed photos, accelerometer reading, sound, and GPS are used to determine information such as road smoothness and traffic density.

While there have been sensors and sensor networks developed for measuring thermal comfort, there has been few works that have actuated temperatures using participatory sensing. There are many advantages to utilizing participatory sensing in this application. Psychological studies have also shown that direct control can lead to greater satisfaction [Pac90], which is an important factor for building management to reduce user complaints. Using actual user input rather than a model for the conditioning feedback loop is likely to lead to better quality of service.

The authors of [Jaz11] developed a participatory sensing cell phone application for measuring temperature, lighting and air quality. For measuring temperature, they used a 5 point scale rather than a 7 point scale. They tested their application over the course of 10 days in 8 different rooms and collected 200 data points; they do not mention how many occupants participated. They compare the measured temperature readings with user data and the analyze differences. They do not, however, use this data to actuate temperature changes. In our deployment, our sensed data is used for temperature actuation and analysis, not solely monitoring. Our work discusses how differences from the measured PMV and sensed data can be used for correction of temperatures. Our deployment has run for a substantially longer period (24 months). Since user feedback directly impacts their environment, the incentive to participate provides us with more data. As illustration, a single room of our deployment over a 10 day period had 362 data points whereas their 8 room deployment only yielded 200 data points.

In [MJK] departures from the standard 7 point scale and instead measures occupant comfort on a scale between -50 (too hot) to +50 (too cold) using a cell phone application. They then use this data to learn occupant comfort levels using the online learning policy UCB1 [ACF02] and LLR [GKJ12]. They treat this problem as a multi-armed bandit problem of picking the correct sequence of temperatures over time in order to maximize thermal comfort and rewards are in the form of favorable occupant feedback. An issue not addressed is the inconsistency of occupant feedback. We found that occupants often do not provide feedback when comfortable. Thus, this scheme may be optimizing for occupants who may not necessarily be present. Another issue not addressed is whether people are truly able to differentiate on a 100 point scale; it is questionable whether or not a person can consistently differentiate between for example +45 and +46.

In the next Chapter, we start discussion of our work by first examining strategies for occupancy prediction that can be used for accuation strategies for conditioning.

CHAPTER 3

Occupancy Modeling

As mentioned in the previous chapters, there are four components we utilize together in order to improve HVAC efficiency; occupancy monitoring, occupancy prediction, use-based HVAC control, and occupancy driven thermal comfort. The initial direction of our research focused on developing a model for occupancy prediction since the prediction models required for control HVAC are largely independent of occupancy monitoring used and we had access and prior experience with an occupancy monitoring system. The goal of this chapter is to develop occupancy models that can predict occupancy of rooms 30-60 minutes into the future that can be used for predictive HVAC control.

In order to predict occupancy, we must have training data and input into the prediction model. For training data, we assume that an occupancy monitoring system is available. For inputs, we assume that occupancy is correlated by both spatially and temporally. With regards to spatial correlation, intuitively occupancy in a room is likely correlated with the occupancy of other nearby rooms. For instance, if a hallway is occupied at one point in time, an adjoining room will likely be occupied in the near future when the occupant exits the hallway to enter the room. Since the occupancies of rooms are correlated, an input into the prediction model will be the current occupancies of rooms. Occupancy is also intuitively correlated to the time of day. For example, we expect at lunch for occupancy of offices to decrease and break-rooms to increase. Thus, time is will also be an input into the prediction model.

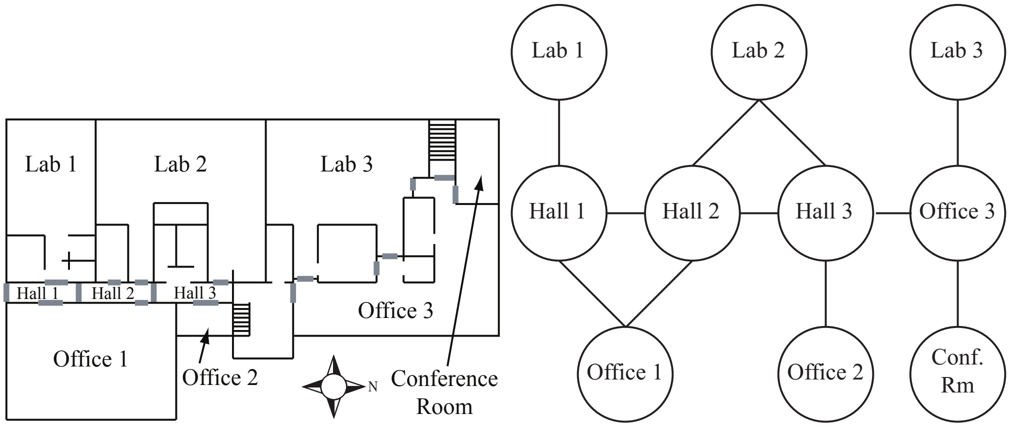


Figure 3.1: The left shows the ten areas data was collected for and the 18 boundaries (gray lines) defining the areas. The right shows a graph representation.

This chapter contributes occupancy modeling techniques that uses inter-room relationships over time to predict occupancy. Our models are developed using accurate real world data rather than occupancy walk-through surveys or other similar estimates of occupancy.

3.1 Data Collection

For the development our models, we used ground truth data; for a live real-world deployment, data from an occupant monitoring system would be used. Five days (Mon-Fri) of ground truth occupancy data were gathered for ten areas using seven cameras covering 18 transition boundaries. Figure 3.1 shows the building areas used for data collection. The cameras recorded images at 1.5 fps. Simple background subtraction methods were used to first help identify images containing people and then processed by hand to verify direction and the boundaries being crossed.

The hall occupancies are generally 0. When the hallway is occupied, the occupancy is usually low and the occupants typically exit within 4 to 8 seconds. There are a few instances that the hall occupancy deviate from this typical pattern.

When the data was collected, there was construction work being done near Lab 1. On Monday and Thursday, about 7 people conducted building inspections in the hallway. There are also some instances where people held lengthy conversations in the hallway sections.

Figure 3.2 shows the occupancy data for ten areas. The Office 1, Office 3, Lab 1, and Lab 3 show occupancy pattens more typical of work routines though with some slight variations. Office 1 is used by school administration. Workers come in consistently at 8am, leave for lunch, come back from lunch, and then leave at 5pm. Office 3, Lab 1, and Lab 3 are used by professors and graduate students. Though the same general daily pattern can be seen, we can also see some occupants stay later (all night in some cases) and arrive and leave at odd hours. Office 2 and the Conference Room are similar since they both are mainly used for meetings. Office 2 is a small office that is used for student counseling meetings and certain administrative work. Lab 2 is currently not being used by any department and serves as storage space.

3.2 Multivariate Gaussian Model

In this section, we discuss a simple occupancy model that utilizes multiple Gaussian fits over the data. In particular, we focus on constructing a model that can simulate the occupancy for two specific areas that are represented by the ground truth data. Given its simplicity, this model serves as a coarse baseline model for other occupancy simulation and prediction approaches.

On an intuitive level, over the course of a day we expect occupancy to increase in the morning when people arrive for work, decrease when people go to lunch, increase when people return from lunch, and then eventually drop to zero when people leave for the day. These are the general increases and decreases of occupancy we can expect based on our real world experiences. The strategy behind

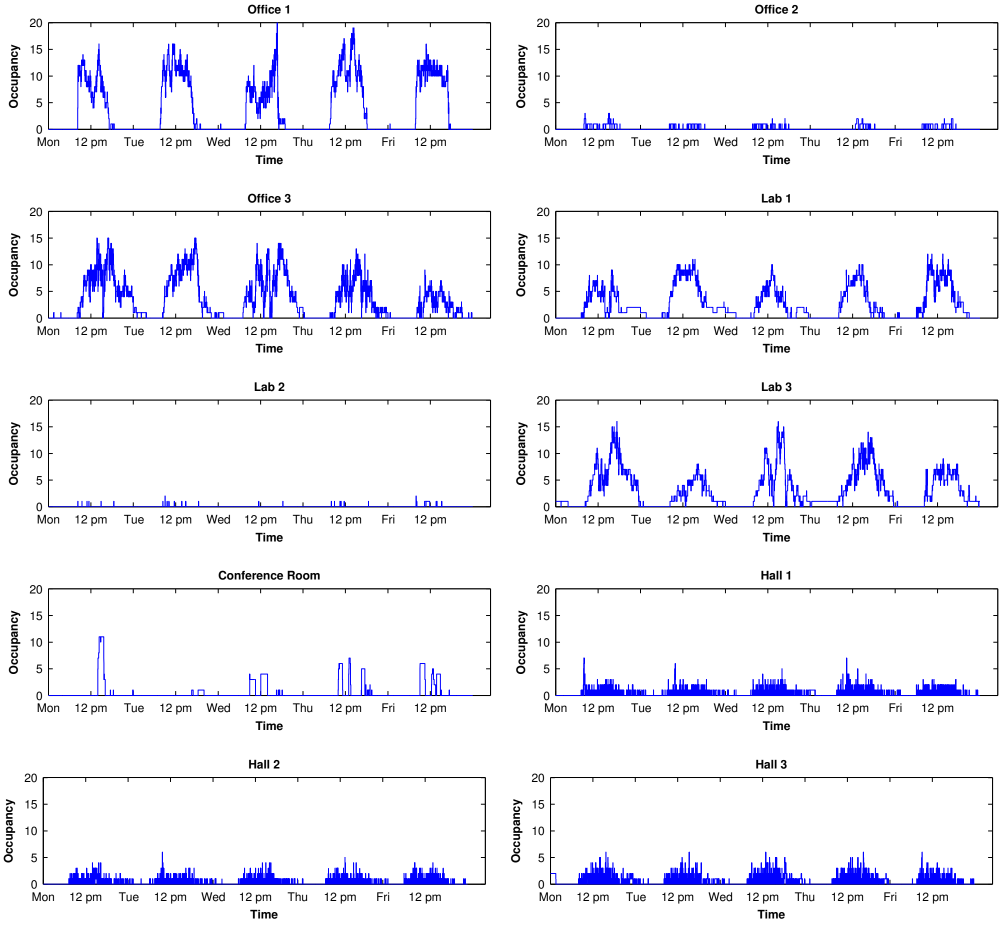


Figure 3.2: 5 days of ground truth occupancy data for eight areas.

this approach is to model each of the increases and decreases separately. However, as the occupancy data suggests in Figure 3.2, there may be other regular phenomenon affecting occupancy other than those based on intuition.

We start first by defining the occupancy state for a building. The occupancy state x is a vector of m dimensions, where x_i is a nonnegative integer corresponding to the occupancy (number of people) in room i, and a probability distribution $p(x)$. Over a specified period of time, the room occupancies are normally distributed, and can thus be represented with a multivariate Gaussian model. For our model, we define hourly pdfs that allow the calculation of the probability for a particular occupancy state occurring within a given hour.

Let $O = (r_1 \dots r_m)$ denote all occupancies that occur per second during a period of time and r_i is a vector of occupancies for room i where $i = 1 \dots m$. Let μ_i denote the average occupancy for room r_i over the time period. We calculate a vector of means $\mu = (\mu_1, \dots, \mu_m)$ and covariance matrix Σ from O. Using μ and Σ, we define a probability density function (PDF) f:

$$f(O; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left\{ \frac{1}{2} (O - \mu)^T \Sigma^{-1} (O - \mu) \right\}$$

We define hourly Gaussian models with mean μ_h and covariance matrix Σ_h where f can give a probability of an occupancy occurring for a specific dataset O_h for hour h.

Using this function, we can randomly draw occupancy vectors from the distribution. Given a starting occupancy vector, all the possible occupancies that can occur in the next timestep are examined. For each possible occupancy vector, the probability of the occupancy occurring using the current PDF is calculated.

A major drawback of this method is that it erroneously predicts a great deal of pacing behavior. For example, if the model predicts a person leaves the office to enter a hallway at one time-step, the model shows a high probability that the person will re-enter the office in the next time-step. This occurs since the distribution does not take into account the behavior observed in the previous time-step. This can cause predictions to favor parts of the distribution that have high probabilities. This is particularly pronounced for rooms rarely occupied. In these cases, the model rarely allows room entry and vacates the room too quickly. Since intended purpose of the model is HVAC actuation, utilization of this model would lead to rooms to be to under or over conditioned.

While the model considers both the temporal and inter-room correlations, it does not take into account previous behavior and does not accurately capture the usage of seldom used rooms. In the following section, we propose a more

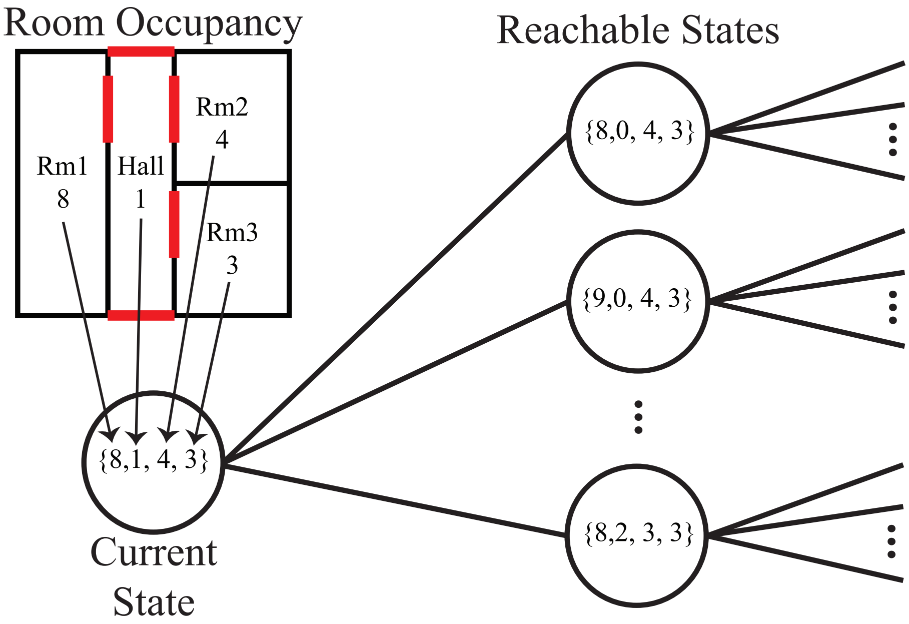


Figure 3.3: The occupancy state representation.

sophisticated technique using a Markov Chain approach in order to better capture the dynamics of the rooms. In particular, we wish to better capture the use of seldom occupied rooms. In order to create a prediction model suitable for HVAC control, we need a model that captures the temporal nature of the occupancy changes along with the inter-room correlations and occupant usage of the areas.

3.3 Markov Chain Model

In this section, we model the temporal dynamics of the occupancy in a building with a Markov Chain (MC). The state of the chain consists of the occupancy at each room and transitions to a new state occurs with a probability that depends only on the current state and the time. This allows us to predict the occupancy distribution at $t + \Delta t$ given the occupancy distribution at time t by multiplying the O_t with the transition matrix. This allows us to predict when rooms will be likely occupied and begin conditioning beforehand.

Note that state transitions must respect the room graph of the building as shown in Figure 3.1. Internal movement of occupants among adjacent rooms maintains the sum of the components of the occupancy vector. External movement of occupants to/from the outside of the building from rooms connected to the outside will change the said sum. This limits the states to which we can transition to from a given state (we call these *reachable* states). Figure 3.3 illustrates this.

The MC state at each time is represented by a vector where each component represents occupancy in a specific room (Figure 3.3). Let R represent the set of n rooms to be modeled. For each room in R, there is a maximum occupancy. We define $X = \{x_0, ..., x_m\}$ to be the set of all room occupancy combinations that are possible given the maximum room occupancies of R where m is the total number of states. Thus, X represents all observable occupancy states that can be represented by a given set of rooms R.

One issue that needs to be addressed is the potentially large state space. Assume we have $n = 4$ rooms, each with a maximum occupancy of 20 people. In this case, the MC will contain $160,000$ states. As the number of states increases, more data is required to calculate the probabilities of the observable states. The transition matrix is sparse because many transitions are impossible due to physical constraints of the building. However, the fact still remains that as more rooms are added, the number of states to manage increases exponentially. To reduce the component cost, we define the MC only on the states that were actually observed during training. Although we will not be able to generate transitions to all observable states, practically our model will be realistic if using a large enough training set. As the WSN gathers data over time, more of the state space will be represented. Additional strategies of reducing the state space are addressed in Section 3.3.4.

With the observable states of the MC defined, we next define the transition probability matrix. Let p_{ij} represent the probability of moving from state j to i where x_t represents an occupancy state at time t. For each state in X we calculate

$$p_{ij} = P(x_{t+1} = i | x_t = j).$$

Since the training data collected is at the resolution of seconds, each time step of the MC represents 1 second. The transition probabilities are estimated from the data normalized counts:

$$p_{ij} = n_{ij} / \sum_{k=1}^{m} n_{ik},$$

where n_{ij} is the number of times a transition from state i to state j in the set, and m is the total number of states.

Since certain occupancy changes occur with greater probability depending on the time of day, the time of day must be incorporated into the model. Consider a person standing in a hallway at 8:00 am. Since it is early in the day, it is likely that the person has arrived for work and will move into either a lab or office. However, if we consider the same scenario at 8:00 pm, it is more likely that the person will exit the hallway in order to leave the building. To incorporate time into the model, we define multiple transition matrices that govern the state changes within different slots of time, thus defining an inhomogeneous MC. That is, we consider separate models for each hour that will be trained using the data for that hour, just as we did in Section 3.2 with the Gaussian model.

While considering only observed (as opposed to observable) states and multiple transition matrices does allow us to model occupancy dynamics efficiently, a problem arises. Since we only consider states observed in the training data, partitioning the data in temporal sets will create discontinuities at the slot boundaries. Suppose we partition the data to create hourly transition matrices and are predicting occupancy for hours h and $h + 1$. After 3600 steps (one hour), we are in some state x, the hour changes from h to $h + 1$, and the model switches to the hourly transition matrix for $h + 1$. It is possible the transition matrix for hour

$h+1$ has no probability for occupancy state x. This occurs if state X never occurs in the training data for hour $h+1$. Even though in reality the state x can occur in hour $h+1$, if x does not occur in hour $h+1$ of the training data, then we cannot calculate the transition probabilities for x. This cannot be solved by introducing a small epsilon probability value of transitioning into another state because the next state chosen may also not be represented in the transition matrix. The MC becomes a random walk until it enters a state that was captured by the training data.

Similarly sink states can also occur if an occupancy state only occurs once in the training set. Suppose we have a set of occupancy states O that we use to train a transition matrix. Let x be the last occupancy state of the training set of O. If x is a unique state in O, then x never transitions to any other state. If the occupancy state transitions to x, it will remain in that state until the model changes to a transition matrix that does contain a transition probability. We propose two different methods of solving the boundary discontinuities and avoiding sink states.

3.3.1 Closest Distance Markov Chain

The closest distance Markov Chain (CDMC) attempts to solve the discontinuities at the time slot boundaries. Suppose we finish simulating hour h and are currently in state x. Before switching to the transition matrix for $h+1$, we first check to see if it contains a probability for x. If the probability does not exist, then we examine all the states of transition matrix $h+1$ that does have a probability and choose the state closest to x.

We define the distance between two states to be the number of transition to or from the outside world in order to account for the difference between the states. Let M_i represent the minimum number of area transitions that is needed for a

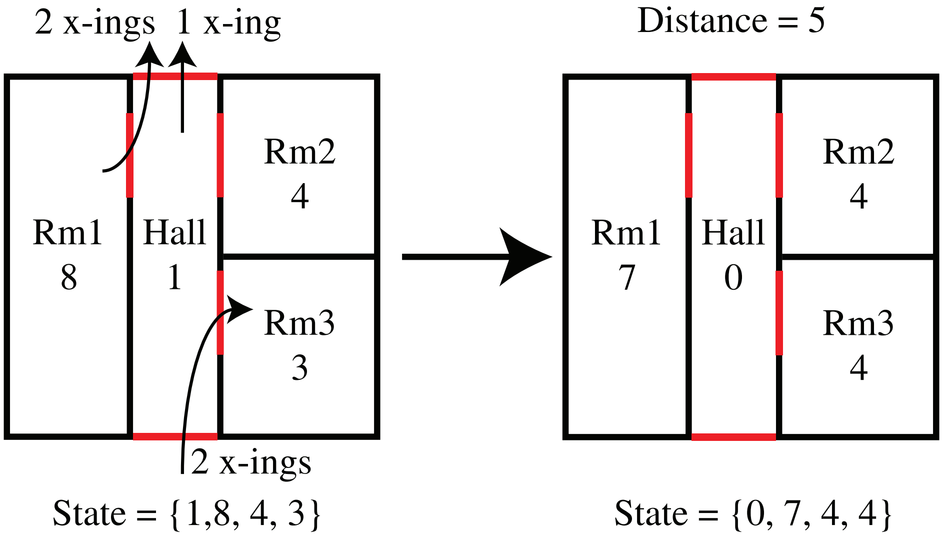


Figure 3.4: Example of the closest distance metric with state $(Hall, Rm1, Rm2, Rm3)$ and distance 5.

single person to enter or exit out of a building from room i. Let $A = (a_0, ..., a_n)$ and $B = (b_0, ..., b_n)$ represent two occupancy states where a_i and b_i are the occupancies of room i. We define distance $d_{A,B}$ between A and B to be $d_{A,B} = \sum_{i=0}^{n} M_i|a_i - b_i|$. Figure 3.4 shows an example of this distance metric. For this four room example, the distance between the states is 5. One transition boundary is crossed to leave Hall. Two transitions crossings are needed for the person to leave Rm1. Two transition crossings are needed for the person to enter Rm3. Rm1 and Rm2 require two transitions each since a person must first enter the hallway and then from the hallway exit to the outside world.

Although this method eliminates sink states at the hourly breaks, it does not guarantee that the sequence of states is valid. It is possible for people to "teleport" in or out of areas. Consider the floor plan in Figure 3.4. Suppose we have the state $(0, 0, 0, 0)$ (all the rooms are empty) and the closest state is $(0, 0, 2, 0)$ (two people are in the office). The transition from $(0, 0, 0, 0)$ to $(0, 0, 2, 0)$ is impossible since the two people must first pass through the hallway to enter the office. Also, it is still possible to enter sink state in the middle of a prediction. Applying the closest distance metric again does little to help since it will likely choose the previous state and then immediately re-enter the sink state.

3.3.2 Moving Window Markov Chain

Sink states only occur when the next transition matrix does not contain a probability for the current state. To avoid sink states, we wish to ensure that the current state exists in the next transition matrix. One way to avoid sink states is to use a transition matrix that is trained over a moving window of data. As state changes progress, the window of data that the transition matrix is trained changes.

Thus far, we have defined the MC model to have hourly transition matrices that capture the occupancy behavior for a particular hour. However, the choice to train over a specific hour is somewhat arbitrary. We can create a transition matrix for any specified window of time. Rather than defining a transition matrix every hour, we instead define a transition matrix every n minutes with each transition matrix still trained with an hour of training data. Thus, each transition matrix overlaps its neighboring transition matrices by $60 - n$ minutes. By having this overlap, we can reduce the likelihood of entering a sink state. Let D represent the training data. Let the function $g(D, t_0, t_1)$ return the transition matrix trained with D defined for the window of time t_0 to t_1. For example, $g(D$, 8:00 am, 9:00 am) represents a transition matrix trained with D for hour 8:00 am to 9:00 am. If $n = 10$ minutes, then we get the following 140 overlapping transition matrices,

$$\begin{aligned}
T_1 &= g(D,\ \text{00:00},\ \text{1:00}) \\
T_2 &= g(D,\ \text{00:10},\ \text{1:10}) \\
&\vdots \\
T_{140} &= g(D,\ \text{23:00},\ \text{00:00})
\end{aligned}$$

Once the transition matrices are defined, we perform predictions with Algorithm 1.

Algorithm 1 MWMC

```
$CurrState \leftarrow$Starting state
$TransMat \leftarrow T_i$ Starting transition matrix
$Sim \leftarrow \{\}$
$w \leftarrow 0$ Window move counter
for $t = 1$ to Simulation Duration do
  if $w > n * 60$ seconds then
    $NextMat \leftarrow T_{i+1}$ Next transition matrix
    if $CurState \in NextMat$ then
      $TransMat \leftarrow NextMat$
      $w \leftarrow 0$ Reset window counter
      $i \leftarrow i + 1$
    end if
  end if

  if $w < 2 * n * 60$ seconds then
    $CurrState \leftarrow RandomState(TransMat, CurrState)$
  else
    $CurrState \leftarrow$Closest State in $T_{i+2}$
  end if
  $Sim.add(CurrState)$
  $w \leftarrow w + 1$
end for
```

Once entering a transition matrix, after n*60 steps (n minutes) have past, the algorithm attempts to move to the next overlapping transition matrix. However, it only changes transition matrices if the currently defined state exists in the next transition matrix. Otherwise it remains in the current transition matrix. Though this lessens the probability of entering a sink state it is still possible to encounter one. If this occurs, we use closest state of the next transition matrix.

After exploring the parameters by trial and error, we choose $n = 30$ in order to reduce the number of overlapping matrices while still reducing the possibility of entering a sink state.

3.3.3 Blended Markov Chain

The blended Markov chain approach solves the problem of encountering a sink state without using the closest distance method. In this method, we blend each matrix into a single state space containing each state of each transition matrix. This is similar to the MWMC, but considers all states observed from the training data.

If the day is partitioned in K parts, then we have K transition matrices $T_1 \dots T_K$ each with m states. We now linearly combine these K transition matrices to obtain K blended transition matrices $\overline{T}_1, \dots, \overline{T}_K$, as follows. The blended transition matrix for slot t is

$$\overline{T}_t = \sum_{s=1}^{K} \beta_{ts} T_s \tag{3.1}$$

where the coefficients $\beta_{t1}, \dots, \beta_{tK}$ are positive and sum to one (so that $\overline{T}_t$ is a valid transition matrix).

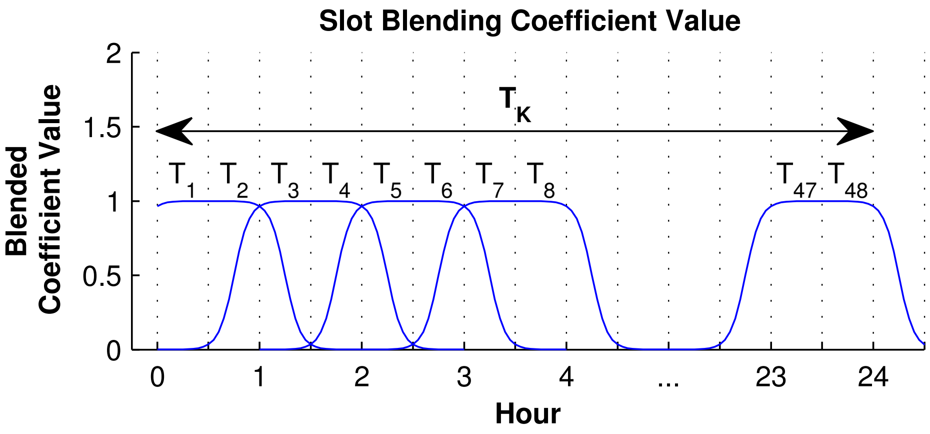


Figure 3.5: Relationship of blending coefficient, hourly time slot, and transition matrices.

We want these coefficients to be approximately 1 for close slots and quickly decrease to 0 for farther slots. We can achieve this by defining the coefficients as

$$\beta_{ts} = \frac{\alpha(c_t - c_s)}{\sum_{s'=1}^{K} \alpha(c_t - c_{s'})} \tag{3.2}$$

where c_t, c_s are the centers of slots t, s, and with a “slot” function

$$\alpha(x) = \sigma\left(\frac{2a}{d}\left(x + \frac{d}{2}\right)\right) - \sigma\left(\frac{2a}{d}\left(x - \frac{d}{2}\right)\right) \qquad x \in \mathbb{R} \tag{3.3}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, $a > 0$ is the slope at the slot boundaries and $d > 0$ is the slot width. Since the sigmoid is monotonically increasing and satisfies $\sigma(-x) = 1 - \sigma(x)$, it follows that $\alpha(x)$ is positive and symmetric around its center. Figure 3.5 shows the slot functions for several slots. This way, each entry in the blended transition matrix $\overline{T}_s$ incorporates information from all slots, but heavily weighting slot s.

For our model, we choose $K = 48$, partitioning the data into half hour transition matrices with slot widths of $d = 3$. We set a slope of $a = 10$ at the boundaries. We choose the c_k to be center of the current hour. Figure 3.5 shows the blending coefficient for these particular parameters. By defining the parameters in such a manner, we create overlapping slot boundaries. This increases the number of preferred states available to transition into, and decreases the chance of choosing states completely outside the slot boundaries. The coefficient heavily favors transitions to states within a given hour time slot and somewhat considers states in the adjacent half-hour slots. States outside this time frame are still considered, but with greatly reduced probability.

These parameters were chosen through trial and error. The criteria of the parameter selection was to maximize slot size while minimizing the transitions into states completely outside the preferred slot boundaries. While larger values of K could be used, smaller values of K are preferred since a large K reduces the slot size and the number of observed states for each transition matrix. We also prefer to only draw from states that are close with respect to time.

3.3.4 Scalability to Large Buildings

The number of observable states increases exponentially with the number of rooms, but as described earlier we limit the complexity of our model by using only states observed in the data sample. If this has length s over a given time period, then the number of observed states N satisfies $N \leq s$, and practically $N \ll s$ because of repeated states. Besides, the $N \times N$ transition matrix T is typically sparse, because physical constrains make many transitions impossible and because not every possible transition is observed in the sample anyway. Indeed, the number of nonzeros in T must be smaller than s, and again is practically much smaller than s because of repeated transitions. For our 5-day training set, we observed $s = 432$K samples resulting in $N = 3\,809$ and $19\,578$ nonzero transitions ($= 0.14\%$ of all N^2 transitions).

It is conceivable that, with a very large number of rooms and a very long

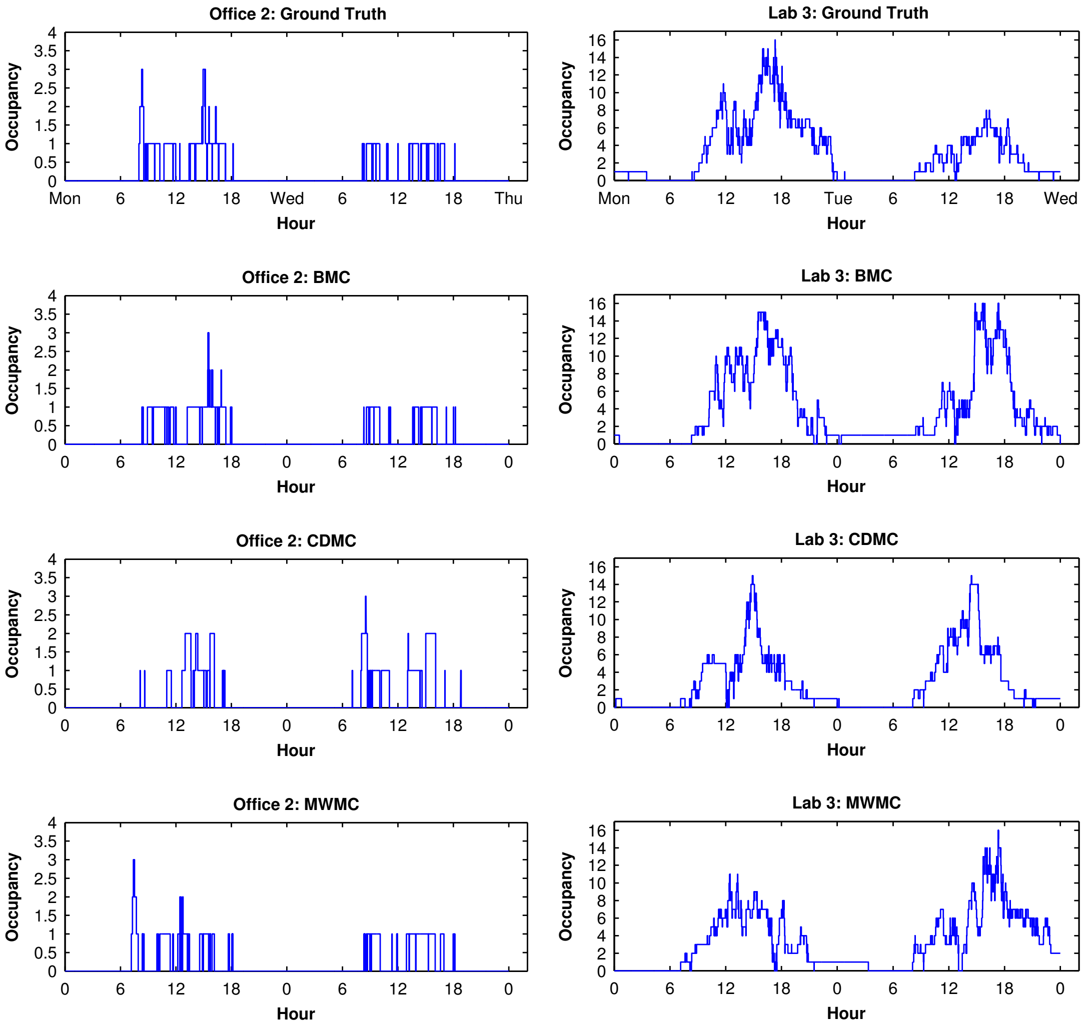


Figure 3.6: Comparison of training data with the models for several areas.

sample, N or the number of nonzeros of T are prohibitively large. Several simple strategies can be used to achieve a manageable model. Firstly, s should not be larger than necessary to achieve a sufficiently accurate model. Another strategy for managing the number of states is to utilize multiple MC models. Suppose we have a building with 40 rooms that is divided into two wings each containing 20 rooms. Rather than having a state representation that includes all 40 rooms, we can train one MC per wing. The trade off is that more MC models would be required to describe all possible relationships of all the rooms. Lastly, it is often possible to aggregate multiple rooms together thus reducing the number of states. This is because HVAC systems typically condition groups of rooms called zones. We can aggregate room occupancies within a zone, and then create an MC model using zones rather than individual rooms.

3.4 Markov Chain Performance

In this section, we will examine how well BMC, CDMC, and MWMC capture room occupancy. A total of 7 days of ground truth data was collected. Five days (Mon-Fri) of this ground truth occupancy data were used to train the models. For each model, 100 simulations of 24 hours were created. The remaining three days (Mon, Wed, Fri) of ground truth data are used for testing. Figure 3.6 shows two days of training data along with model simulated room occupancy schedules. Even without formal tests, the models seems to be capturing the dynamics of the ground truth room occupancies. As expected, the CDMC enters sink states and remains in the state until the end of the hour. Qualitatively, BMC and MWMC seem to model occupancy variability better than CDMC.

3.4.1 Comparison Metrics

We use three metrics to evaluate the quantitative performance of the models. We first examine how long room occupancies remain static to measure occupancy variability. We expect our occupancy models to remain at the different levels of occupancy for similar durations as compared with the ground truth data. The next metric utilizes Jensen-Shannon divergence (JSD). This is a method that applies Kullback-Leibler divergence (KLD) to compare the similarity of two distributions. The advantage of JSD over KLD is that JSD will always return a finite value and is symmetrical. We compare the room occupancy distributions of the models and testing data for different windows of time during the day. The last metric considered is the rate people enter and exit a room. By examining the durations between entrances and exits of a room, we can measure the flow of occupants in and out of a room. Specifically, for a window of time w, we calculate $\lambda_{in,w}$ and $\lambda_{out,w}$ for each room where the variables represent the rate of flow in and the rate of flow out of a room respectively. Time is taken into account since the rate of flow changes depending on the time of day.

3.4.2 Evaluation

Figure 3.7 compares the average static durations of the various occupancy levels with those seen in the testing set. Duration differences closer to 0 indicate a closer fit to testing data. For high traffic areas and rooms with low occupancy, BMC and CDMC show similar differences. CDMC typically shows larger differences of durations since the CDMC still tends to get stuck in states near the ends of the hourly partitions. This causes the occupancy level to remain static for periods of time longer than normal. Areas with larger occupancies are prone to containing sink states as more data is required to cover the possible state space. This can be seen in Figure 3.7 for Office 1 where CDMC remains at several occupancy levels for long durations. Similar results were found for the other areas with higher levels of occupancy such as Office 1, Office 3, Lab 1, and Lab 3. BMC and MWMC show very similar differences. This is expected as both of these methods have a similar method of favoring states within a frame of time. However, as mentioned

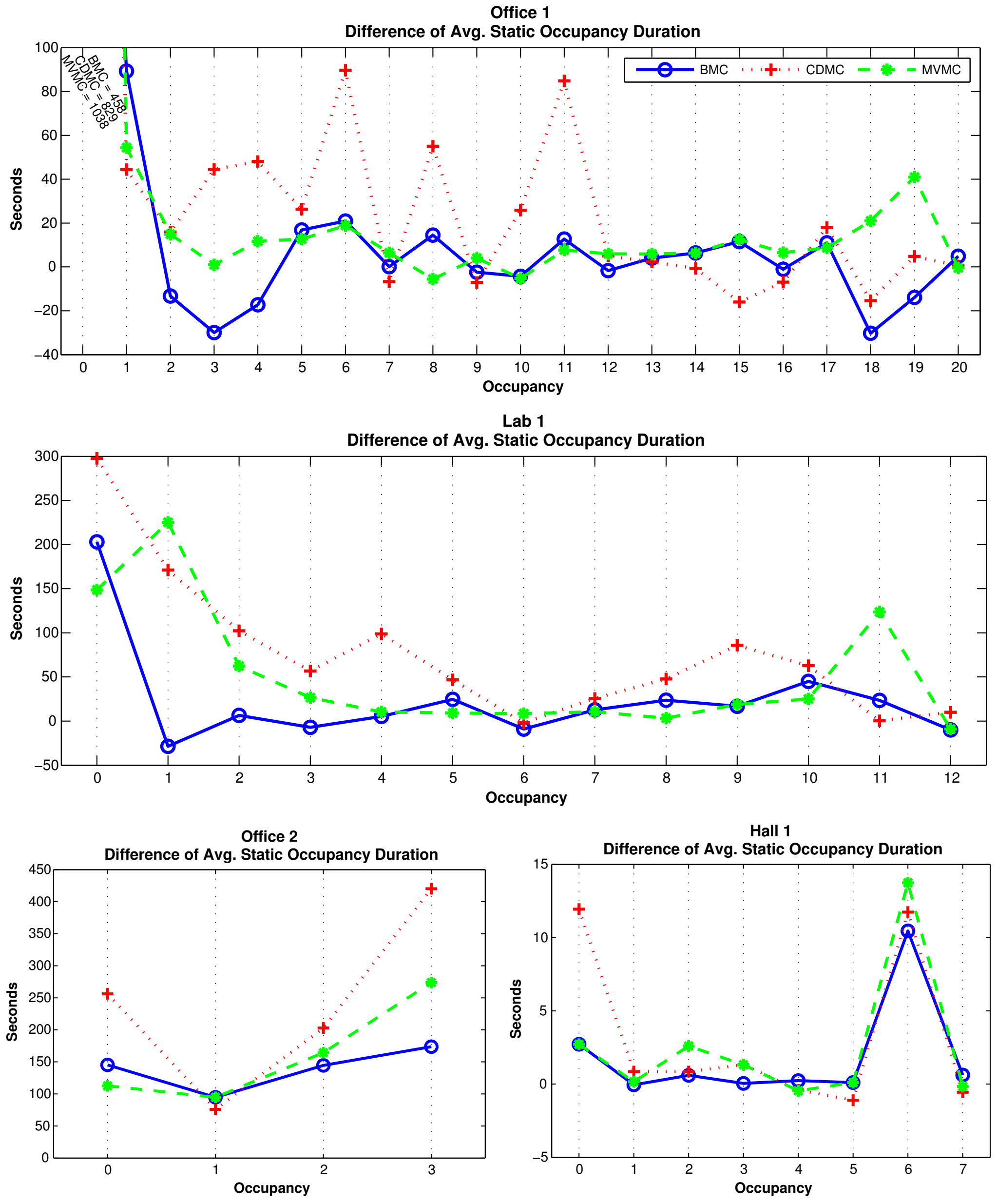


Figure 3.7: The difference of the average static occupancy duration from 7am - 10pm.

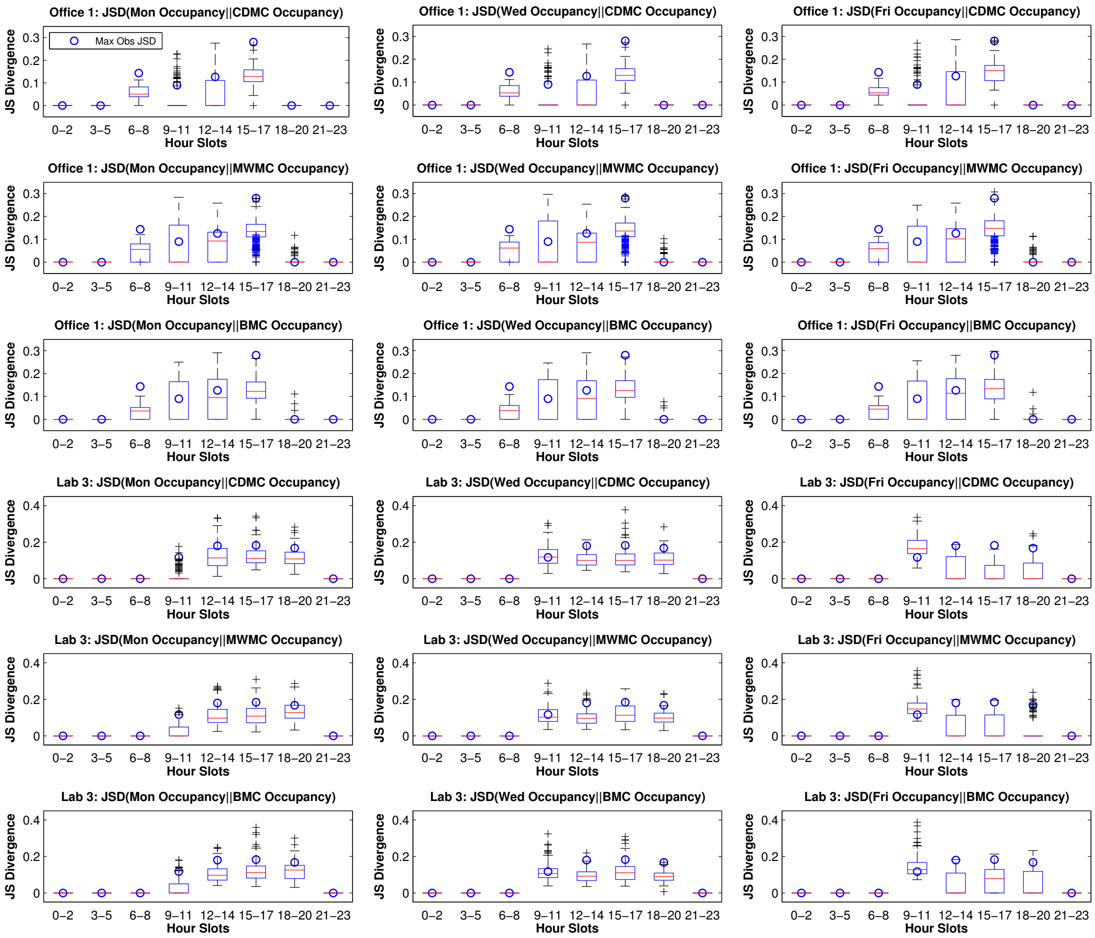


Figure 3.8: Each boxplot is for the JSDs observed within the time slot where the + is an outlier.

previously, the MWMC still has the possibility of entering a sink state causing static occupancy levels. However, this occurs much less frequently when compared to the CDMC.

We next examine the JSD for the different models. We partition the day into 3 hour slots and examine the JSD for each window of time. Examining the ground truth data in Figure 3.2, we can expect a fair amount of variation among different days. To establish a baseline of how much divergence is typical from day to day, we compare the occupancy distributions of each testing set day to the remaining testing set days for each window of time and establish the maximum divergence

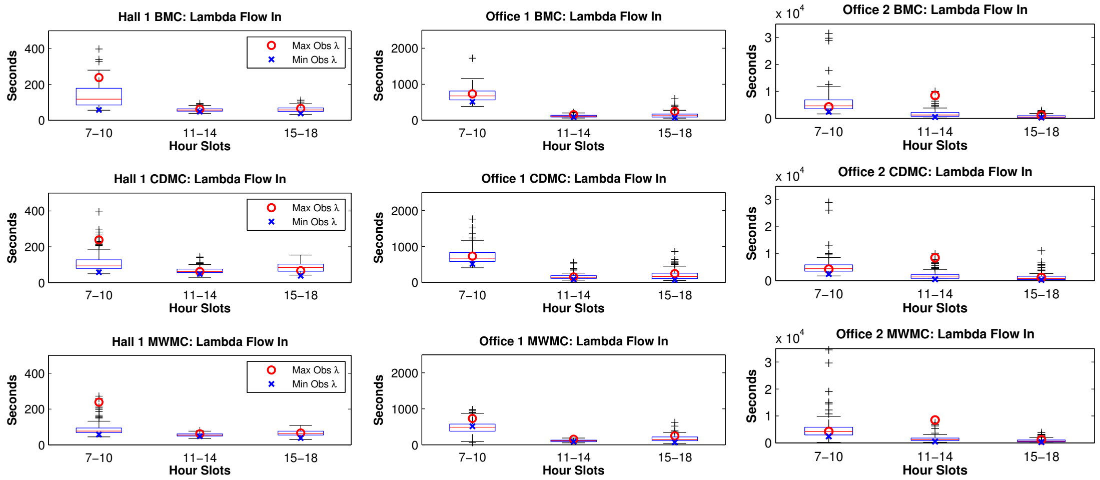


Figure 3.9: Flows for Hall 1, Office 1, and Office 2 and the min and max flows observed in the testing set.

for each time slot. Rooms that are mostly empty such as Lab2 and the conference room have JSD of nearly 0 for each window of time. Figure 3.8 shows the JSD for the rooms that are consistently occupied. When we compare the testing set for each model, we see that the JSD for each window is consistent. We find that all three models have JSD's that are below the maximum observed JSD. We also see that MWMC and BMC have very similar results, which is expected as both strategies have a similar mechanism for choosing the next state.

Lastly we examine the flow of occupants for each rooms. We consider the hours 7am - 10am to be morning, 11pm - 2pm to be afternoon, and 3pm - 6pm to be evening. Based on these windows of time, we calculate the flow into and out of rooms. Figure 3.9 shows the flows into Hall 1 and Office 1 for the different simulations. As with JSD, we will use the ground truth test data to establish upper and lower limits for the λ values. Again, all three methods fall mostly within the limits. However, CDMC and MWMC tends to have more outliers, which again is caused by sink states. However, since sink states occur with less frequency with the MWMC, the MWMC has fewer outliers than CDMC.

Overall, all three models perform well, but in general, BMC performs better since it does not encounter discontinuities at the boundaries, which can occur for both CDMC and MWMC. For this particular training set, the MWMC did not exhibit discontinuities at the slot boundaries; however it is possible with other datasets this could occur. The sequence predicted by BMC is also always valid, which may not be the case for the CDMC and MWMC where the closest distance is used to escape sink states.

3.5 Summary

In this chapter, we develop and examine occupancy prediction models for the predictive conditioning component of our system. Based on intuition backed by ground truth data, we use inter-room occupancy correlation and time of data as inputs into our model. Our initial model that used Gaussian fits to poorly predict occupancy since it was not able to accurately predict areas that are only briefly occupied. We next propose a statistical model of the temporal occupancy of a building based on an inhomogeneous Markov chain. Since it is not practical to consider every possibility state of room occupancy, We control the complexity of the state space by using only states observed during training, and prevent discontinuities in the chain by a closest distance Markov chain (CDMC) approach, or by continuously blending the transition matrices over time (BMC). Comparing samples generated from the model with ground-truth data using various metrics shows that the model captures the occupancy dynamics accurately. While all the Markov based models would be suitable for many occupancy prediction applications, the BMC has advantages over the other models for predictive conditioning. In particular, the BMC avoids discontinuities at the temporal bins. While the other models may work the majority of the time, these discontinuities can potentially lead to “stuck” occupied states, which should be avoided for pre-conditioning

HVAC strategies.

With the occupancy prediction model developed, we next examine how to measure occupancy. In the next two chapters, we will discuss two different approaches for monitoring occupancy using wireless sensor networks. With an occupant estimation system, we are able to control ventilation in real-time and train an occupancy model such as the BMC for predictive control.

CHAPTER 4

OPTNet: OPtical Turnstile Network

In the previous Chapter, we developed multiple occupancy prediction models that would be suitable for determining the likelihood a room will be occupied in the near future. The next system of the four main components is the occupancy monitoring system.

As described in Chapter 1, an occupancy estimation system can be used to adjust HVAC system based on actual usage. In particular, it allows both temperature and ventilation to be controlled. In addition to HVAC control, it also can supply the training data required for the occupancy prediction model used for predictive HVAC control of room temperature.

We developed two systems for real-time occupancy monitoring; a camera based system deployed in public hallways that captures occupant movement among rooms and a thermal array based solution deployed directly in rooms. In this chapter, we examine our first developed system utilizing cameras deployed in public hallways along with motion sensor in rooms to infer occupancy. The camera system was developed before the more effective and inexpensive thermal sensing arrays used in Chapter 5 became available.

4.1 Inferring Occupancy

By using cameras as optical turnstiles, it is possible to infer occupancy levels. Privacy issues are avoided by only deploying in public hallways where security cameras are already present. Moreover, by performing local video processing on the cameras, we avoid the transmission of raw feeds to a central location and we prevent invasion of privacy. Furthermore, this distributed solution enhances scalability and significantly reduces network load, improving system lifetime in the bandwidth limited camera sensor network. Existing optical counting systems exist, but they are difficult to retrofit to buildings and require hard wiring [tru]. For our solution, we wish to utilize motes that can be easily deployed in even older buildings. With efficient use motes, battery replacement can be done along with routine light-bulb replacement.

4.2 Motivation

Though there are several methods that are commonly used for detecting occupancy within modern buildings, these methods present limitations. Passive infrared (PIR) sensors, commonly used for controlling lighting, are a simple way of detecting if a room is occupied or not, but do not give information regarding how many people occupy the room. This information is necessary for CO_2 ventilation. Using CO_2 sensors directly for regulating CO_2 is also unsuitable for conditioning strategies. CO_2 buildup is slow, and by the time sensors detect high levels of CO_2 that trigger ventilation, occupants of the room are likely to already feel uncomfortable [FFS06]. If not properly calibrated, these sensors can also be inaccurate [FFS06]. Electrical loads have also been used for occupancy estimation [AC01]. However, this method assumes each occupant contributes to the load and requires accurate occupancy data for calibration.

Unlike lighting, the thermal ramp up or down of a room involves delay. While an optical system of occupancy monitoring can give occupancy in near real-time, reactively conditioning a room will likely leave occupants uncomfortable until target temperatures are met. In order to ensure occupant comfort, we must be able to predict when occupants are likely to enter a room and begin conditioning beforehand. The length of the prediction is based on how long it takes for the room to reach temperature from a target temperature. We achieve this by using a blended Markov Chain as described in Chapter 3.

In this chapter we contribute the following:

- We developed OPTNet, an occupancy estimation system comprised of 22 node wireless camera nodes, and BONet, a 40 node PIR wireless sensor network. We show how lightweight on-board image processing algorithms along with classification techniques can be used in order to accurately detect occupants' transitions.
- We show how errors in occupancy sensing can be corrected by fusing data from an occupancy transition model together with sensor data using a particle filter.

4.3 Overview

In this chapter we develop OPTNet, a wireless camera sensor network and BONet, a PIR sensor network. A particle filter that fuses the sensing data from OPTNet and BONet with the output of an occupancy transition model in order to better estimate the current occupancy in each room is also developed. Figure 4.1 shows all system components.

4.4 OPTNet

OPtical **T**urnstile **Net**work (OPTNet) is a new low power wireless camera system that is able to accurately monitor room occupancy in near real-time. We start with a description of the hardware and deployment in Section 4.4.1 followed by

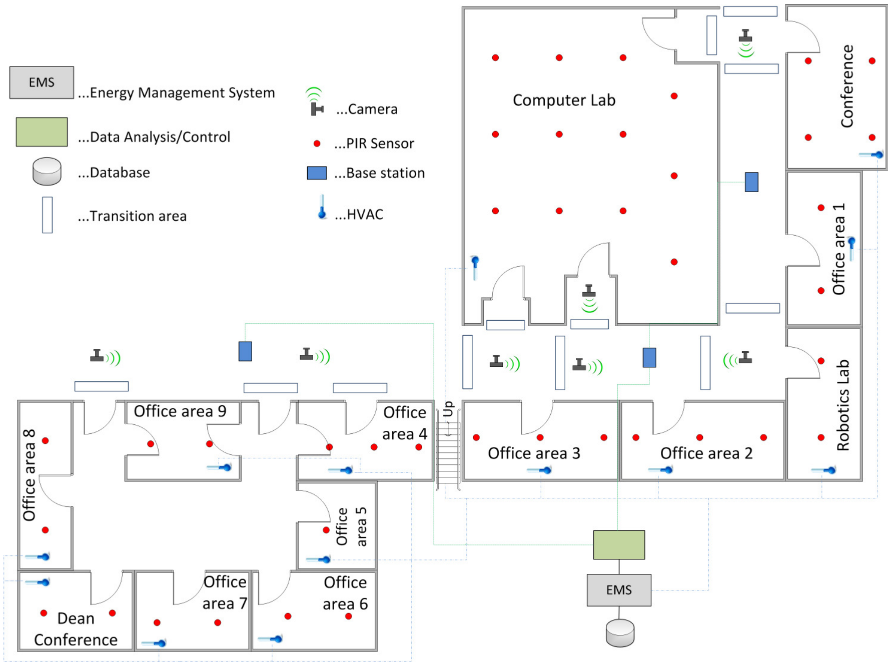


Figure 4.1: System components

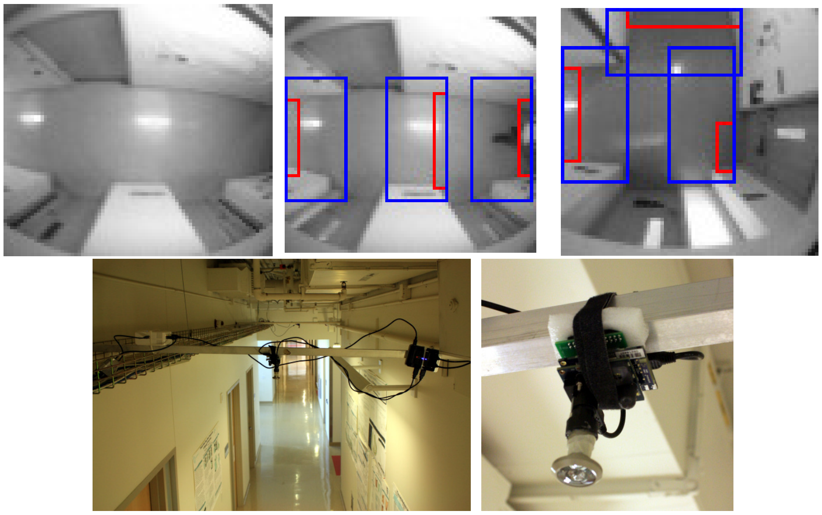


Figure 4.2: Transition areas (bigger blue boxes) and trigger areas (small red boxes).

a discussion of sensor placement in Section 4.4.2. In Section 4.4.3, we describe a lightweight image processing method that can be used with a classification algorithm to detect transitions.

4.4.1 Overview and Design Challenges

We use the Imote2 platform developed by Intel and the IMB400 Camera [imb] developed by Xbow. The mote uses an XScale processor currently set to run at 208 MHz. The mote has 64MB of memory and a CC2420 radio chip. The camera can capture 32 fps at 640x320 resolution. By utilizing a fish-eye lens fashioned from a door peephole (see Figure 4.2), we are able to roughly view 100 ft^2 when the cameras are deployed at a height of 10 ft. Therefore the camera has a viewing angle of about 160 degrees. We have 22 camera nodes deployed on two floors capable of measuring the occupancy of 60 areas. In our test deployment the nodes have a wired energy supply, which makes it easier for us to run different experiments. We also have the option to power a node with an 11 Ah battery pack which provides enough energy to run a node for about 45 days.

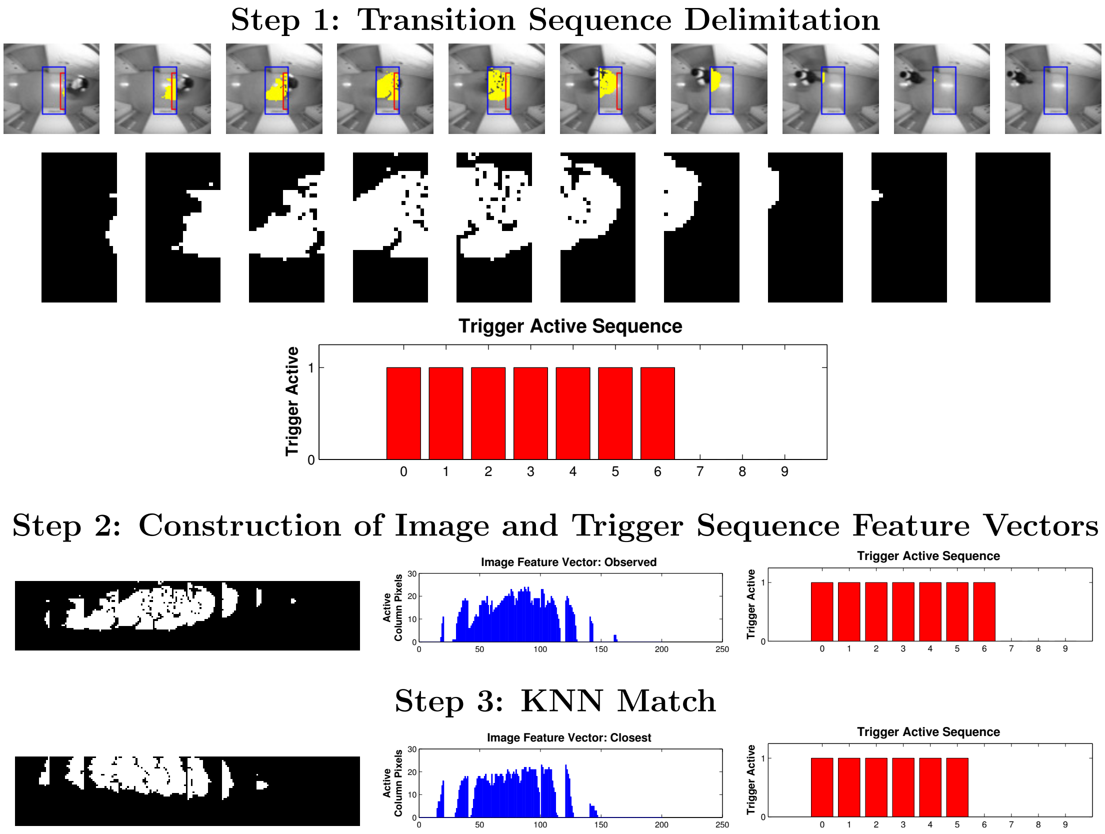
The goal of the system is accurate detection of occupants moving between areas. In order for the control system to be useful, accurate occupancy detection, particularly for empty rooms, is critical. Empty rooms provide an opportunity to let room temperatures float to more energy efficient levels. Also important is the ventilation of the room, which is proportional to occupancy.

There are several design challenges. Motes have limited resources available. Though the IMB400 cameras are able to capture 32 fps, the Imote2 radio is only able to transmit $\approx$ 100 kbps under typical conditions, making real time image streaming infeasible. This bandwidth issue can be partially solved if we only send images of interest and buffer the images until they can be sent. However, this requires heavy use of the radio, which is one of the most energy consuming components. Instead, it is more efficient to extract only useful features from the images and transmit a small amount of information. Thus, our design challenge is to design an efficient on-board image processing algorithm that can compress data to be processed at the base-station.

Occupancy tracking is still an open problem in the computer vision community. Many tracking algorithms focus on the person, typically correlating the identity of a person between frames using feature-based algorithms such as SIFT [Low99] and SURF [BET08]. While it is possible to run these algorithms on the Imote2, they are computationally expensive. Also, placing cameras directly into rooms raises serious privacy concerns. OPTNet thus only uses cameras installed along hallways. In our application, we only care about occupant transitions between areas, not the identity or precise location of specific occupants. Instead, we view this problem as a motion recognition problem, where we are attempting to recognize a sequence of images as a transition.

4.4.2 Camera Placement

HVAC systems are designed to condition zones within the building, where each zone is comprised of one or more rooms and is controlled to a specific temperature/ventilation. Figure 4.1 shows the zones for our deployment. As previously mentioned, we are attempting to capture the transitions that occur between zones in order to track the occupancy of an HVAC zone; the cameras operate as optical turnstiles. In order to capture all transitions, cameras must be deployed in such a way that all the entrances and exits are covered for a zone. Thus, the distribution and number of cameras depends on the entry/exit points of an area. More specifically, there must be one camera per entrance for a zone. For example, to track all the transitions for Offices area 1 (see Figure 4.1), a camera is placed in front of the Conference area and another in front of the Robotics lab. Since zones share entry/exit points, a camera can serve multiple areas; the camera deployed



Step 1: Transition Sequence Delimitation

Step 2: Construction of Image and Trigger Sequence Feature Vectors

Step 3: KNN Match

Figure 4.3: Above is a summary of transition classification procedure. Previous image background subtraction detects movement within a transition area. The image sequence is then concatenated to form a larger matrix shown as a picture on the right of Step 2. A feature vector is then constructed where each position corresponds to the number of active pixels in each column of the matrix (left Step 2). This vector is then classified using KNN against known labeled data (Step 3).

in front of the Conference area is used to track transitions from the Computer Lab, Conference area, and Office area 1.

4.4.3 Transition Detection

In this section, we describe a fast lightweight image processing algorithm that can be used for transition detection. Our strategy is to classify a sequence of images as a specific type of transition. We accomplish this by first detecting motion within a target transition area using background subtraction. We then determine the start and end of a transition from a continuous sequence of images containing motion. This sequence is then transformed into a vector that can be sent back to the base-station to be compared against known labeled data using the k-Nearest Neighbors (KNN) algorithm.

4.4.3.1 Motion Detection

For human motion detection we use previous frame background subtraction with a static threshold [FP02]. We tried several other background subtraction methods (e.g. moving average, weighted moving average), but we found previous frame subtraction to work the best for our application. In particular, it is robust against rapid lighting changes caused by opened doors as it can quickly reduce the number of erroneously active pixels. This reduces the number of perceived transition sequences caused by people loitering near or within transition areas. It also quickly removes objects from images such as boxes and chairs, which occur frequently in the public hallways that our deployment is located. Also advantageous is that it requires minimal resources to run in real-time on the mote.

4.4.3.2 Delimiting a Transition Sequence

The first problem we address is identifying a sequence of images that could potentially contain a transition. We start by defining a transition area, which is a small 1x2 m^2 area on the ground typically in front of a door or within a hallway. Figure 4.2 (left) shows an example of a transition area. In addition to a transition area, we also include a trigger area (smaller inner red boxes in Figure 4.2) that is placed on one side of the transition area. The trigger area helps to distinguish when someone actually crosses the area and the direction of travel (Figure 4.4). Figure 4.3 shows a transition example. If a transition area has been empty for 3 previous frames ($\approx$ 0.6 seconds), when a person walks into the transition area, this signals the beginning of a transition sequence. Detection is achieved using previous frame background subtraction using a static threshold. A sequence is ended if there is no activity close to the trigger area (60 cm) for 3 consecutive frames. Transitions are assumed to take a minimum of 0.25 seconds. With the non-active frames buffering the front or back of the image sequence, about 0.25 seconds in duration, we assume any sequence shorter than 5 frames ($\approx$ 1.0 seconds) is not a valid transition.

We found this method of delimiting sequences works well. The previous frame background subtraction, in particular, is useful for discarding spurious transition sequences. People loitering within the transition areas are usually relatively stationary. Movement will activate the sequence, but since they are not moving significantly, they quickly disappear as part of the background causing the sequence to end quickly. A loiterer typically generates multiple sequences less than the minimum 5 frames. Sudden flashes of light are adapted to quickly and the perceived sequences are typically shorter than the minimum 5 frame sequence.

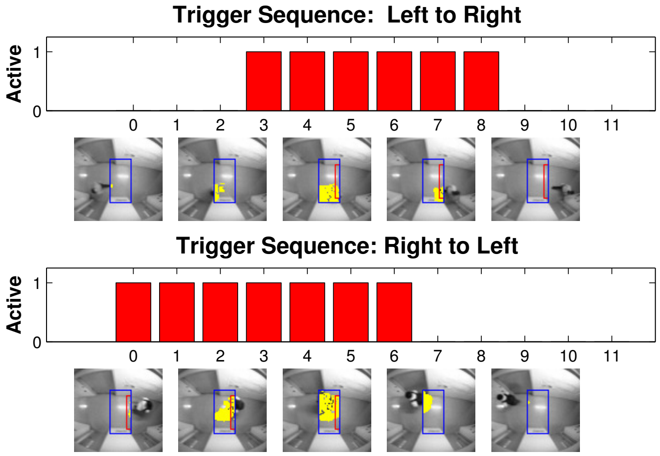


Figure 4.4: The sequence of the trigger state helps indicate direction. The red box indicates the trigger state is active.

4.4.3.3 Sequence Feature Vectors

Next, we define how to construct a feature vector from an image sequence, which will then be used with a KNN classifier to determine the best classification. Let $[f_0 \dots f_n]$ denote a set of transition sequence frames, each f_i encoded as a binary matrix of size $w \times h$ where w and h are the width and height respectively. Thus, $f_0 \dots f_n$ denotes the complete image transition sequence. Let c_i^j denote the ith column vector of f_j. We define the one dimensional image feature vector v where $v_i = \sum c_i^j$. By summing the columns we are removing the y axis locality of the person moving through the frame. A person transitioning at the top, middle, or bottom of a hallway will thus generate a similar image feature vector. The number of people transitioning is also still retained. If two people walk side by side, the magnitude of the column vector will be roughly doubled.

In addition to creating a feature vector of the image sequence, we also create a feature vector of the trigger state. For each frame of the image sequence, we determine if the trigger area is active or not. The state of the trigger area is defined to be active when a certain threshold of pixels (5% for our deployment) differs from the corresponding background. For each frame of the transition sequence we determine the state of this trigger. A vector of the trigger states is incorporated into the KNN classification to determine the direction of the transition. Figure 4.4 shows how the sequence of the trigger state can be used to infer direction. In this example, a left to right sequence will have non-active trigger states at the beginning and end. A right to left sequence will have active trigger states only at the beginning.

While it would be possible to use the entire transition area as the feature vector, this would require more data to be sent via radio. If we assume a 10 image sequence, then each binary 20×40 transition area needs to be sent along with 10 bits for the trigger sequence. This would be $(20 \cdot 40) \cdot 10 + 10 = 8,010$ bits. If we were to send the feature vectors, we would only require $(20 \cdot 6) \cdot 10 + 10 = 1,210$ bits or ≈ 152 bytes for an entire sequence (6 bits per column, 10 bits for the trigger sequence). This can roughly put in 2 packets using the node RF transceiver mentioned in Section 4.4.1. Further compression could be achieved by aggregating groups of columns. Other techniques could also be potentially used to reduce the size of packets. Learning the dimensionality reduction could be used to preserve class information. This would reduce the size of the feature vector and potentially increase the accuracy of the classification. This along with other compression techniques are left for future work.

K-nearest neighbors is an effective classification technique when few labeled examples are available. For gesture recognition, this has been used for classifying specific gestures [KB]. With the feature vectors defined, we next define the distance metric between feature vectors.

Let $X = (x_1 ... x_m)$ and $Y = (y_1 ... y_n)$ be image feature vectors and $Xt = (xt_1 ... xt_m)$ and $Yt = (yt_1 ... yt_n)$ be the corresponding trigger feature vectors. If $m = n$, the distance between transition sequences is defined as follows,

$$d(X, Y, Xt, Yt) = \sum_i |x_i - y_i| + \alpha \sum_i |xt_i - yt_i| \tag{4.1}$$

where α is a weighting coefficient. This coefficient helps to favor examples with the correct direction. One issue with classification is the varying lengths of the transition sequences. Transition sequence length depends on how many people cross a transition area at walking speed. Rather than match only with transitions of equal length, we also consider sequences within ±20%. Dynamic time warping is also another possible strategy that we save for future work [MR81]. If $m > n$ then the distance is defined as,

$$d(X, Y, Xt, Yt) = \underset{off}{\operatorname{argmin}} \sum_{i=off}^{m-off} |x_i - y_i| + \alpha \sum_{i=off}^{m-off} |xt_i - yt_i| \tag{4.2}$$

where off is an offset. In other words, we are finding the minimum distance when the smaller vector subtracted from a subset of the larger vector at different offsets. Next we define how we choose from the k nearest neighbors. Let $M = \{l_0 \cdots l_k\}$ be the set of k closest labeled matches and $D = \{d_0 \cdots d_k\}$ be the corresponding distances. Rather than classify based on the most frequent label of M, we weight each label l_i by $1/d_i$ and choose the label with the greatest summed weight.

4.4.4 OPTNet Evaluation

With the image algorithm defined, we next evaluate the accuracy of the algorithm. For each boundary, we gathered 48 hours of ground truth transition sequence data. This data was gathered by manually examining the original images seen by OPTNet cameras and hand labeling each transition sequence. The first 24 hours of ground truth is used for training. We trained each boundary area with 150 transition sequences taken from different periods of the day. The remaining 24 hours of ground truth is used for testing. For the weighting coefficient, we set

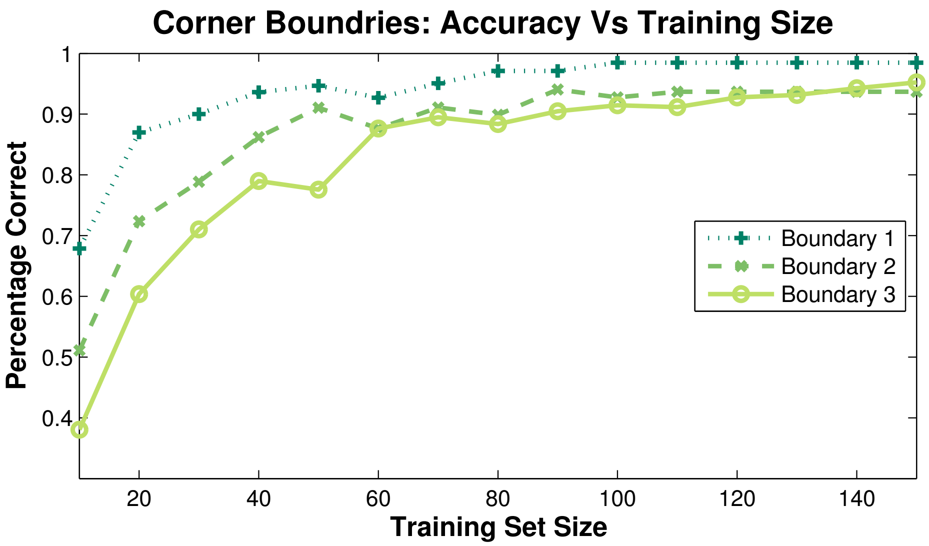


Figure 4.5: Overall accuracy over 24 hours as a function of training set size.

$\alpha = 10$ after trying several weighting factors. While it would be preferable to gather data and test the system for different times of the year to examine the performance, we leave this as future work given the amount of time and effort required to gather ground truth data. We explored many different methods to help assist the gathering of ground truth data. Initially we used a state of the art technique that can count torsos and legs [Ope, VJ01, LM02] to identify images that contain people but found this was only able to identify 80% of the images with people. Even the most advanced techniques are not completely accurate and need manual correction. We used a simple previous background subtraction technique to identify images that contain moving objects and manually processed these images.

In this section, we examine the directional and overall accuracy. We define directional accuracy as number of transitions where the direction is classified correctly divided by the total number of transitions. We define overall accuracy as the number of transitions where the direction and number of occupants are classified correctly divided by the total number of transitions. Figure 4.6 shows the

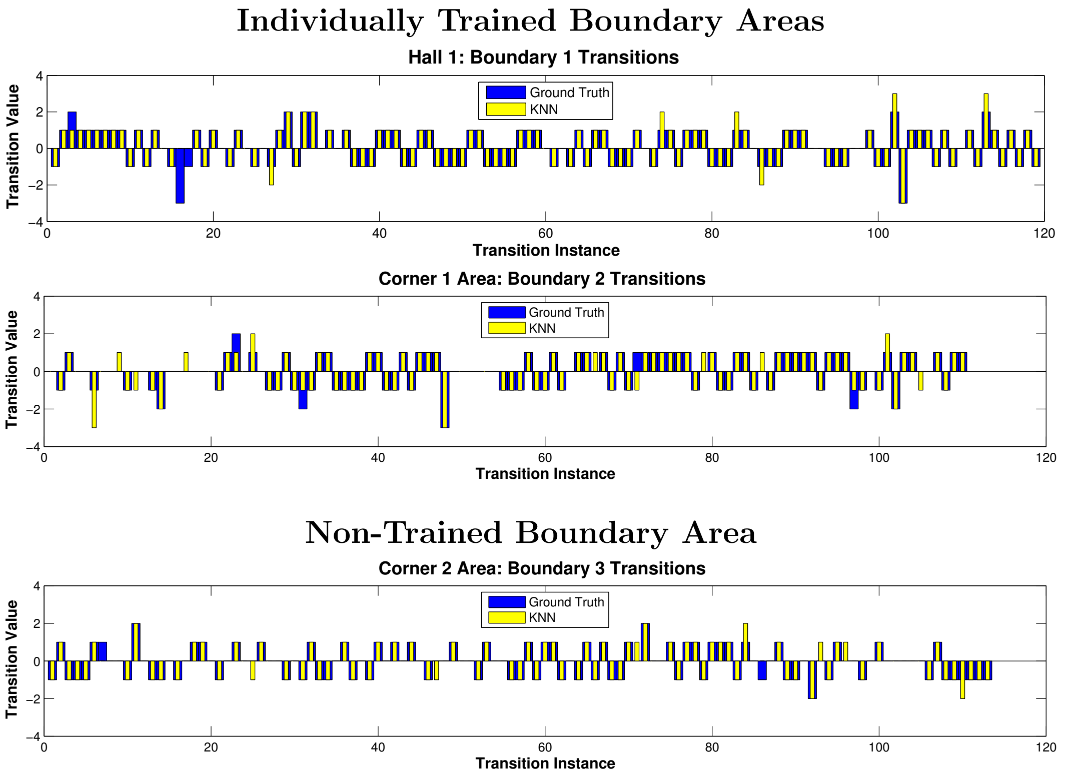


Figure 4.6: Ground Truth compared with KNN model for two different transition areas.

transitions of two different boundary areas over the period of 24 hours. In both cases, we see that the classification of the KNN model is very close to the ground truth. In particular, we see that the directional error is very small. In the case of Hall 1, the directional accuracy was 98.3% and the overall accuracy was 92.4%. Corner 1 had 8 incorrect directions (5 false positive, 3 false negative) out of 110 detected transitions (92.7%) and an overall accuracy of 87.3%. For both cases, 150 training examples were used. We also found that KNN classifier had similar performance when applied to *different* transition areas using training data from *different* areas than the ones being tested. Figure 4.6 shows the accuracy when applying the classifier for a different transition area using 300 training examples. In this case we have a directional accuracy of 94.0% and an overall accuracy of 93.8%.

From a deployment perspective, a classifier can be substantially trained at a few transition areas and then applied to other boundaries without requiring excessive data gathering for training. Figure 4.5 shows the relationship between training set size and accuracy for Hall 1. We took a random subset of transitions from the training set and examined the overall accuracy over 24 hours. Only 90 samples (roughly 45 each direction) are required to achieve above 90% directional accuracy. We found similar results for the other transition areas. Transition areas near corners or highly variable lighting tended to have lower accuracy. In these cases, additional training data would increase the accuracy as suggested by Figure 4.5. More generally, one wants to position transition areas where people are likely pass through and not loiter. We also found it helpful to choose areas with less changes of lighting; this cannot always be achieved since key transition points near exits must contain a transition area in order to capture the occupancy of areas. We also found that transition areas deployed in hallways performed better than transitions placed directly in front of doorways. This is because there is greater variation for the ways people transition through doorways. We also

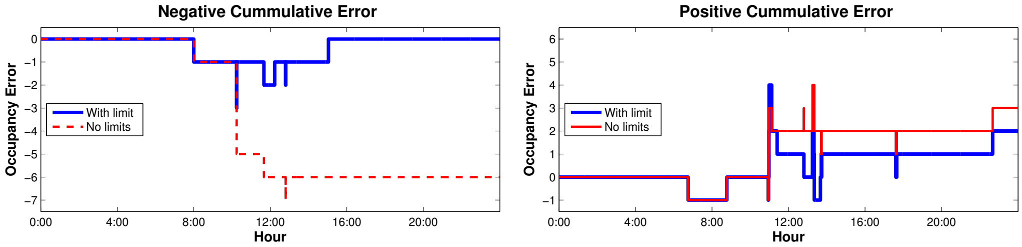


Figure 4.7: Occupancy error over time with/without maximum/minimum occupancy limits.

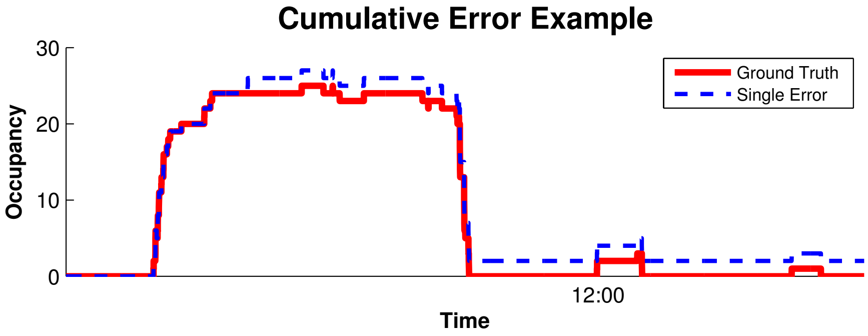


Figure 4.8: A single transition error continues to affect occupancy estimate.

receive more and better quality training examples for transitions through hallways than for transitions directly in front of doors.

4.4.4.1 Occupancy Error

In the previous section, we evaluated the accuracy of the system to detect occupant transitions between areas. Since we plan to use these sensed transitions to infer occupancy, next evaluate the ability of the system to indirectly estimate occupancy. In order to evaluate the error of the system to infer occupancy, we collected 5 days of ground truth occupancy data and compared it to the system estimate of occupancy. The ground truth was collected by examining all the images ($\approx 50,000$) captured by the system and manually annotating transitions between the different areas.

While the individual errors of the transitions are low, the effect of these errors is cumulative with respect to occupancy. Figure 4.8 shows an example where a single error at the beginning of the day causes an over-count of 1 for the rest of the day. Thus, the problem with this technique is with the integration. We employ several basic strategies in order to limit this type of error. If a transition area has a particular bias, maximum and minimum occupancies limit the magnitude of the cumulative bias. Anytime we obtain negative occupancy we assume the occupancy is 0. Similarly, for each room, we assume a maximum occupancy and do not allow the occupancy estimate to be greater than this limit. Figure 4.8 shows examples of how occupancy limits help reduce the cumulative error. In these plots we show the difference between the occupancy estimate (with and without limits) and the ground truth. For the negative cumulative error plot, we see that with limits in place, the error is only -1 for the majority of the time, whereas with no limits, the error is -6 for most of time. Similarly, we see in the positive cumulative error plot that the limited occupancy has an error of 1 most of the time; with no limits, the occupancy error is typically 2. The last strategy we use to reduce cumulative error is assuming occupancy is 0 for all rooms at 4am. Though cumulative error is eliminated eventually (i.e zeroing occupancies at 4am), there may be long periods of time where the occupancy provided by the system deviates greatly from the actual occupancy. This can be seen in Figure 4.8. We can see in the negative cumulative error plot that there is an error of -1 for approximately 8 hours. The positive cumulative error plot shows an error of 2 to 3 for nearly 13 hours. In order to improve the occupancy estimate of the system we use a particle filter as described in Section 4.6.

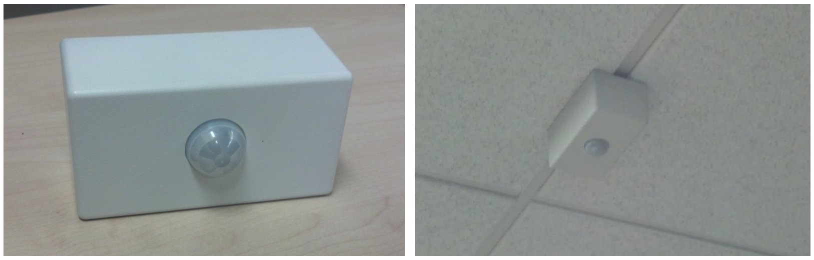


Figure 4.9: PIR node deployed on ceiling

4.5 BONet

Binary **O**ccupancy **Net**work (BONet) is a wireless sensor network of PIR sensors deployed in each office. Each node is comprised of a Tmote and PIR sensor (Figure 4.9). The sensing area of the node is approximately 11 m^2 when deployed at a hight of 3 m. For single offices, a single node provides adequate coverage. For larger areas such as the lab, multiple PIR sensors were used. The PIR sensor is sampled once per second and sends back the total number of triggers every minute. Data was collected using low power listening [T10] and timestamped using Flooding Time Synchronization Protocol [MKS04]. Out of the 34 nodes, 6 experienced false triggering due to calibration issues. However, 3 of these sensor were in the areas where redundant functional PIR sensors were available. Figure 4.10 shows the accuracy for one area. We found the system to have an accuracy of 94% for the 28 functional nodes when compared with ground truth data over a 48 hour period.

4.6 Particle Filter

We implemented a re-sampling based particle filter to achieve a higher accuracy of the room occupancy. This method uses an occupancy prediction model along with the inferred occupancy estimate from the camera system to create a more accurate estimate of occupancy. The particle filter algorithm is a nonparamet-

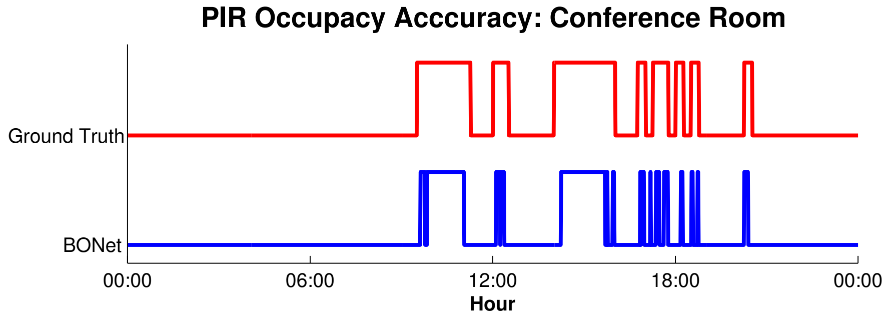


Figure 4.10: PIR accuracy compared with ground truth.

ric implementation of the Bayes filter [TBF06]. We also tested a Kalman filter approach but found the particle filter approach worked better since it is able to track more hypotheses (i.e. future occupancies).

The main concept of the particle filter algorithm is to represent the posterior occupancy state of a certain number of rooms by a set of random state samples drawn from a predefined state transition model and weighted differently according to a measurement model representing the sensor accuracy. These random samples are called particles, and they give a distribution of the estimated posterior state. The particles are defined as:

$$X_t = x_t^1, x_t^2, \ldots x_t^M \tag{4.3}$$

where, each particle x_t^m is a vector representing a set of N rooms with different occupancies at time t. Each position in the vector represents a specific room occupancy; $x_t = [r_1, r_2, \ldots, r_N]$ where r_i is the room occupancy for room i.

Algorithm 2 shows the particle filter algorithm. Data from BONet is used as pre-processing step. If BONet senses a room is unoccupied and OPTNet indicates there is occupancy, then the room occupancy is assumed to be 0; BONet is more reliable than OPTNet for detecting empty rooms. We define OBNet as the combined the OPTNet and BONet system that fuses occupancy data as just described. The particle filter is initialized with the previous particle set X_{t-1} and the current processed sensor output z_t, coming from OBNet. Time t in the particle

Algorithm 2 ParticleFilter(X_{t-1}, z_t):

```
for $m = 1$ to $M$ do
    sample $\hat{x}_t^m$ from TransitionModel($x_{t-1}^m$, $t$)
    add $\hat{x}_t^m$ to $\hat{X}_t$
    for $n = 1$ to $N$ do
        get $w_t^{m,n}$ from Measurement($z_t^n$, $\hat{x}_t^{m,n}$)
    end for
end for
$\bar{w}_t^{m,n}$ = Normalize($w_t^{m,n}$), $m = 1 \dots M, n = 1 \dots N$
for $n = 1$ to $N$ do
    draw $x_t^m$ with probability $\bar{w}_t^{m,n}$ from $\hat{X}_t$, $\forall m \in M$
    add $x_t^m$ to $X_t$
end for
return $X_t$
```

filter represents the current second of a day, starting at 1 second after midnight. M denotes the number of particles in the particle set X_t. In our experiments we tried various numbers of particles between $M = 50$ and $M = 1000$. For the results presented in this paper, we use $M = 100$ particles.

The particle filter algorithm consists of three major steps: 1. sampling from the transition model; 2. calculation of the particle weights; and 3. re-sampling.

The **sampling step** of the particle filter algorithm, M samples are drawn from the transition model. The transition model is represented by a blended Markov Chain as described in [ECC11] and Chapter 3, where each state x_{t-1}^m can reach one possible successor state $\hat{x}_t^m$ with a certain probability $p(\hat{x}_t^m | x_{t-1}^m), m = 1 \dots M$.

In the **weight calculation step** of the particle filter, a specific weight $w_t^{m,n}$ is assigned to each value of each state given the probability of the estimation $\hat{x}_t^{m,n}$ under the sensor measurement z_t^n, for a specific particle m of room n. In our implementation, the weight $w_t^{m,n}$ is determined by the measurement model,

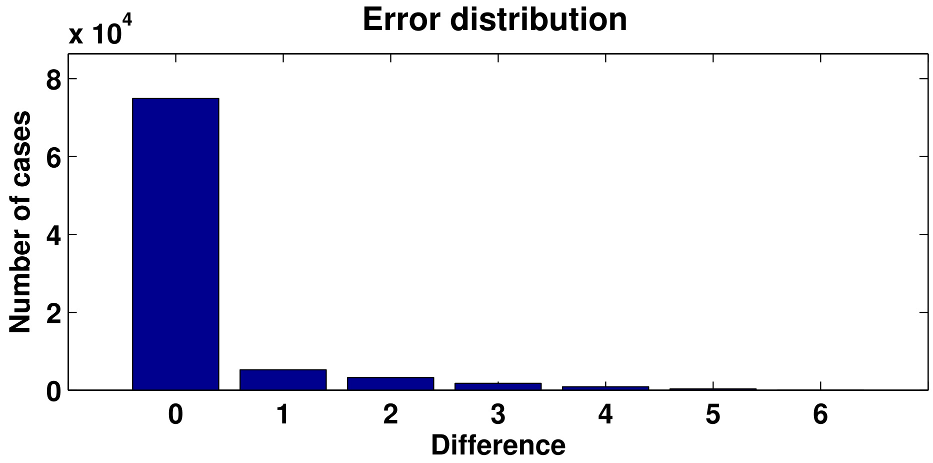


Figure 4.11: Sensor output error distribution.

which is a distribution of the difference between the occupancy ground truth and the occupancy value of the processed sensor output of the training data. The distribution consists of the number of cases of the difference between ground truth and the inferred occupancy estimate from the sensor data as shown in Figure 4.11.

The weight $w_t^{m,n}$ of a specific particle is now determined by calculating the absolute difference between the model estimation $\hat{x}_t^{m,n}$ and the processed sensor output occupancy value z_t^n:

$$x_{tDiff}^{m,n} = \hat{x}_t^{m,n} - z_t^n \tag{4.4}$$

Based on the previously created distribution, the weight $w_t^{m,n}$ of room n in particle m is now determined by dividing the number of cases of a certain difference $x_{tDiff}^{m,n}$ by the complete number of cases. For example, the observed number of cases with a difference of $x_{tDiff}^{m,n} = 1$ between ground truth and processed sensor output is 5,240 out of 86,400 overall cases, which equals 0.06. This is done for every room n within each state m from the transition model.

The **re-sampling step** draws with replacement M samples from the temporary particle set $\hat{X}_t$. Each estimation is drawn with probability weight $w_t^{m,n}$. This method of re-sampling transforms the temporary particle set $\hat{X}_t$ into another par-

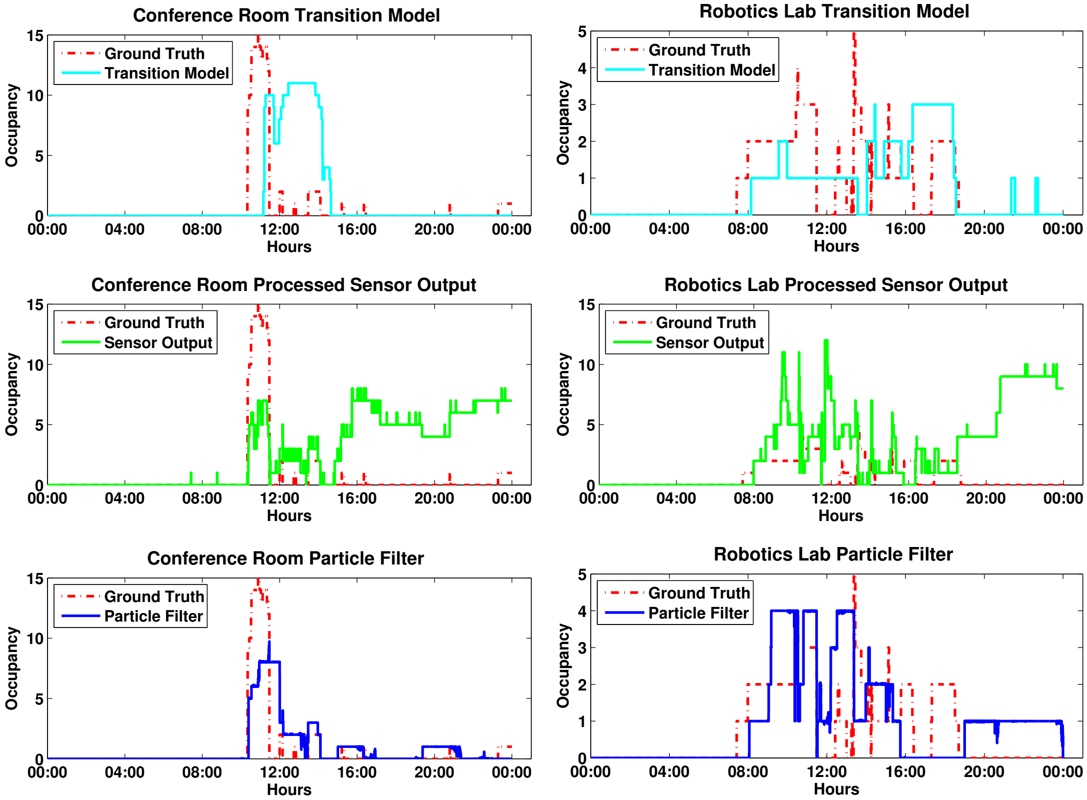


Figure 4.12: Occupancy over time for the transition model, the sensor output and the particle filter for 2 areas

ticle set X_t. By considering the weight of each estimated occupancy value in this step, the result is a different distribution since duplicates of values with a higher weight $w_t^{m,n}$ are more likely to be part of the new distribution than values with a lower weight. The final result is a matrix where each column represents a distribution of the occupancy of a specific room. We obtain the final occupancy state $CurrOcc$ for a number of N rooms by averaging each column over all states of the final particle filter result X_t.

$$CurrOcc = \frac{\sum_{m=1}^{M} x_t^{m,n}}{M}, \forall n \in N \tag{4.5}$$

4.6.1 Particle Filter Results

Figure 4.12 shows the occupancy estimation by a particle filter, the direct sensor output of the OPTNet system. The ground truth shows a meeting in the conference room starting at about 10:30 am and ending at 11:50 am. The direct sensor output of the OPTNet system estimates shows negative occupancy near the beginning of the day, which persists until the meeting starts. Later in the day it overestimates the room occupancy, due to positive cumulative error. This happens between 15:00 and 00:00 when no one is in the room. This error can lead to energy waste if we were to use the direct sensor output of the OPTNet system alone.

The occupancy estimation of the particle filter, which uses the fused OB-Net sensor data and transition model, correctly estimates occupancy at around 10:30 am. Further examination of these short occupancies show a janitor entering/leaving the room. The particle filter tended to show a lower prediction of occupancy than the actual ground truth for the 10:30 am to 11:50 am period. This is because the training data used for the model never experienced a large number of people and the sensor gives a lower occupancy number than the actual ground truth. This can be solved using an extended training set for the transition model.

For the robotics lab, the transition model alone is mostly underestimating the occupancy between 7:30 am and 2:00 pm and only overestimating the occupancy a little bit in the afternoon between 2:30 pm and 6:30 pm. The direct OPTNet sensor output, overestimates the occupancy within the robotics lab almost all day long, especially between 6:30 pm and 00:00 am when the room is actually not occupied. The particle filter overestimates the occupancy only a little bit between 8:00 am and 4:00 pm. Between 4:00 pm and 6:30 pm the particle filter underestimate occupancy of the room. The overestimation of 1 person between 6:30 pm

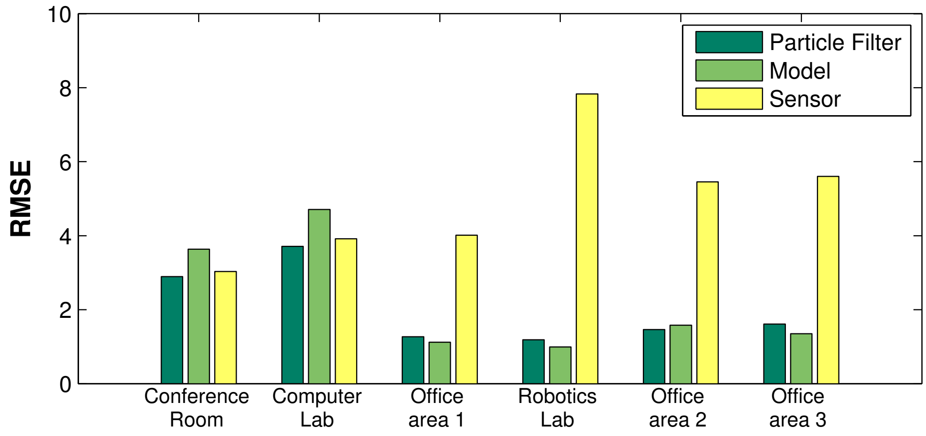


Figure 4.13: Occupancy error over 2 weeks.

and 00:00 am may lead to smaller conditioning costs compared to the sensor, but larger than the model in this time window. Overall, the particle filter estimation in this example provides an occupancy estimation much better than the direct sensor output, and a slightly better estimation than the transition model in this 24 hour example.

We use root mean squared error (RMSE) to evaluate the occupancy estimation accuracy by the transition model, the OBNet sensor data and the particle filter. Figure 4.13 shows the RMSE for 6 different zones. For the conference room and the computer lab, the sensor is more accurate than the model; the sensor output is correcting the estimation of the transition model within the particle filter. For the office areas 1-3 and the robotics lab, the sensor output has a large error caused by transitional bias. The transition model for these rooms shows more accurate results and leads to a slightly better result than the particle filter. Here, the model estimation is correcting the sensor output within the particle filter. Since the error of the sensor is high in the case of office area 1-3 and the computer lab, it is possible that the estimation of the transition model alone is slightly better then the particle filter. The particle filter gave an average RMSE of 1.83 for our building. Overall, the particle filter gave the best results when compared to the model and the sensor.

4.7 Summary

Occupancy measurement is a critical component for controlling HVAC based on occupancy. In this chapter, we design and implement an occupant measurement system using two sensor networks; one network of cameras called OPTNet, and another network of PIR sensors called BONet. We showed that this system is capable of detecting transitions with up to 94% accuracy and that the system is able to generalize to different locations. Combined with BONet and a particle filter, we can bound the error of occupancy within 1.83 people on average for our building.

The choice of cameras for inferring occupancy was based on previous success with this technology and lack of alternative technologies at the time that could estimate occupancy in real-time. In order to address privacy concerns, the cameras had to be placed within hallways and used as turnstiles to indirectly infer occupancy. The main drawback to this technique we found was cumulative error; even a single error had the potential to propagate for a significant amount of time. By using the inferred occupancy from the camera system with an occupancy estimate from an occupancy prediction model in a particle filter, we were able to obtain an improved estimate of occupancy over using OPTNet to inferred occupancy alone. However, this too has several drawbacks. The most significant drawback is the requirement of an occupancy prediction model. In order to create a prediction model, training data is required. Gathering ground truth training data is impractical. Also an issue is that the model will tend to pull occupancy estimates similar to the training data of the occupancy model. OPTNet alone is too inaccurate to infer occupancy alone. While placing cameras directly within offices would remove the problem of cumulative error, this would raise main privacy concerns. While the use of the particle filter occupancy estimates for occupancy based HVAC control is possible, these issues made the system less practical. However, shortly after completing the project, new inexpensive thermal sensing technology became available. With thermal sensing, sensors can be placed directly in the room without the same privacy issues as cameras. Rather than infer occupancy through transitions, thermal sensors can survey a target area and directly measure occupancy, avoiding cumulative error and the need of a particle filter and model. In the next chapter, we show how this thermal sensing technology can be used to estimate occupancy.

CHAPTER 5

ThermoSense: Thermal Based Occupancy Sensing

In the previous chapter, we developed an occupancy detection system that utilized cameras in order to monitor occupants in real-time. As we saw previously, cameras placed in public hallways and function as an "optical" turnstile are prone to cumulative error. If an optical turnstile misses even a single person entering/exiting a space, this error is propagated until another offsetting error occurs or some other mechanism used to correct the error. By applying a particle filter and applying some assumptions, such as assuming the building is empty very early in the morning, we were able to obtain improved estimates of occupancy. While these strategies improved occupancy estimates, it required a model of occupancy in order for the error correction to work. Since the particle filter tended to pull occupancy estimates toward values seen in the training data, the quality of the occupancy estimate required the model to be adequately trained to capture the majority of cases. Ideally, we wish to utilize an occupancy estimation method that is not prone to cumulative error and does not require prior knowledge of occupancy patterns. This became possible when inexpensive thermal sensing became available.

In this chapter, we develop ThermoSense, an occupancy monitoring system that utilizes thermal based sensing and PIR sensors. We develop a novel low-power multi-sensor node for measuring occupancy utilizing a thermal sensor array combined with a PIR sensor. The thermal sensor array used is able to measure temperatures in an 8x8 grid pattern within an 2.5mx2.5m area when deployed on a 3.0m ceiling. We show it is possible to use these temperature readings in order to determine how many people are within the space. Unlike the CO_2 sensor, the thermal array can measure occupancy in near real-time. The thermal array is also not sensitive to optical issues, such as lighting or background changes. By adding a PIR sensor, we increase accuracy of detecting empty spaces and the overall accuracy of the platform. The PIR sensor is also used to reduce the node's power consumption by triggering the mote and thermal array only when someone is present. We test this new platform with a 17-node deployment covering 10 building areas totaling 2,100 sq. ft. for a period of three weeks. In this Chapter, we contribute the following:

- We developed a novel multi-sensor platform for estimating occupancy utilizing a thermal sensor array and a PIR sensor. We performed a full system power consumption analysis and tested a 17-node deployment over a 3 weeks, in 10 HVAC conditioning areas, covering 2,100 sq.ft.

- We developed a new occupancy classification process using the sensor data, which includes thermal map background update, feature extraction, occupancy classifiers and post-processing filtering. We tested three classifiers, including K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), and linear regression (LR). Using this process, we showed that these types of sensors are capable of estimating occupancy with an RMSE of only ≈0.35 persons.

5.1 ThermoSense

We start by describing ThermoSense, a wireless sensor network of nodes that can measure occupancy using a combination of thermal readings and PIR. Figure 5.1 shows the ThermoSense node used for the sensor network.

In order to have an effective wireless occupant measurement system suitable

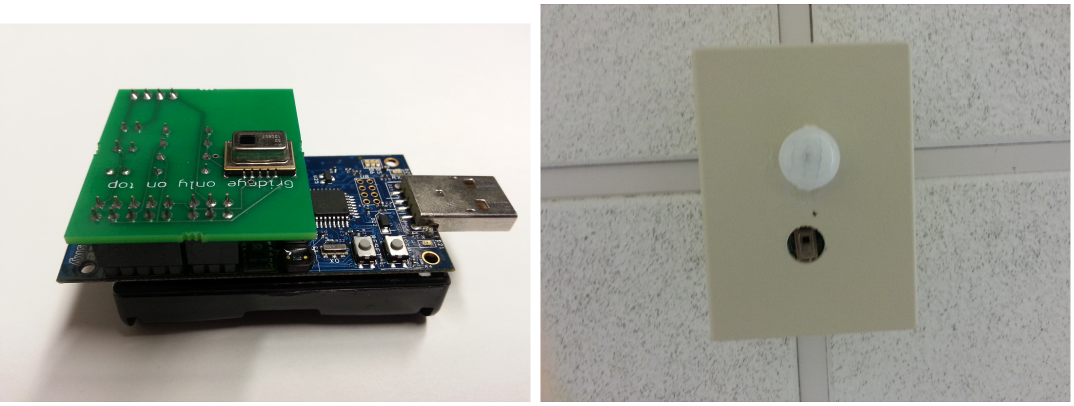


Figure 5.1: Grid-Eye attached to a Tmote (left). Enclosure containing both the Grid-Eye and PIR (right).

for HVAC control, several design considerations must be examined. While wired system for binary occupancy sensing exists, these systems are costly to install and can be difficult to retrofit into older buildings; for our system, we want an easy to deploy wireless system. Since it is wireless and untethered, power consumption is a significant issue. Both the mote and sensors must be low-powered and run in a power efficient manner. For our platform, we make use of a PIR for two reasons. The first is that accurate detection of empty rooms is critical in order to capitalize on potential savings. The second is that the PIR can also be used to sleep components in the system when sensing and communication are not necessary. While the PIR is able to give an accurate and reliable binary indication of occupancy, we need another sensor capable of determining how many occupants are in an area. In conditioned indoor spaces, like the great majority of buildings, occupants are typically warmer than the surroundings, hence a thermal array a thermal sensor array, capable of measuring multiple temperatures within an area, is a viable method for detecting occupancy. Figure 5.2 shows an above view of the area, where each square of the grid is a point of measurement of the thermal sensor array. The shaded red squares shows the points in the grid with higher temperatures, which in this case is the location of the occupant. Multiple

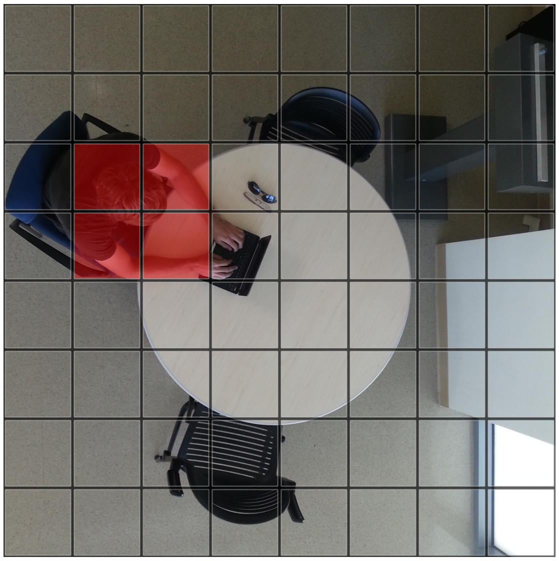


Figure 5.2: 8x8 thermal array sensing an occupant.

occupants can be distinguished by higher density of warmer temperatures.

5.1.1 Hardware

Each ThermoSense node contains a Tmote Sky, PIR, and Grid-Eye Sensor. The Tmote Sky [tmo], produced by MoteIV, has a 8MHz TI-MSP430 micro-controller with 48k Flash storage and 10k RAM. The Tmote communicates using a Chipcon CC2420 radio. The PIR sensor is connected to the Tmote's digital input/output pin while the Grid-Eye is connected to the I2C clock and data pins. We developed a board in order to connect both the Grid-Eye and PIR sensor with the Tmote.

The PIR used in the nodes is the PaPIR EKMB VZ series developed by Panasonic. The PIR is able to detect motion up to 12m with a detection a viewing angle of 102°x92° horizontal by vertically respectively [PaP]. The infrared thermal array used to detect actual occupancy is the Grid-Eye sensor made by Panasonic [gri], priced at $31. This sensor measures 64 temperatures in an 8x8 grid where the physical size of this grid is determined by the distance the sensor is to the target surface. We found the Grid-Eye to be able to sense within a 2.5mx2.5m square

Components Used	Energy Usage
Mote Only (No Duty Cycling)	4.72 mA
Mote + PIR	4.73 mA
Mote + PIR + Grid-Eye	4.76 mA
Mote + PIR + Grid-Eye + Radio	4.82 mA

Table 5.1: Energy Usage for independent components.

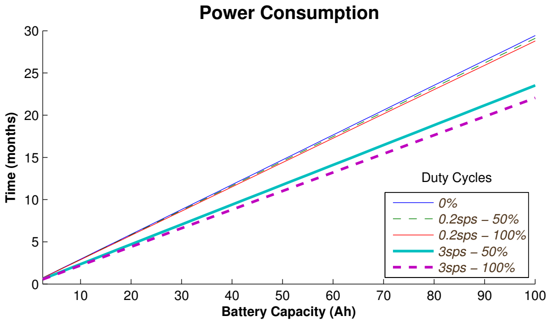


Figure 5.3: Energy usage for different duty cycles.

when placed at a height of 3m. The Grid-Eye measures temperatures from -20C^o to 80C^o with an accuracy of ±2.5C^o and can sample 10 times per second (each sample contains 64 temperature values).

5.1.2 Power Consumption

We used two 3000 mAh lithium batteries for each node in our deployment. We found battery life of ThermoSense node to be over three weeks sampling once every five seconds. For a more detail analysis of the consumption, we measure each component of the platform separately using an oscilloscope. Table 5.1 summarizes the results. The difference in energy between using PIR and Grid-Eye together as compared to the PIR only is small (0.03 mA). Adding the radio with the Grid-Eye and PIR only increases the current use by 0.06 mA. Since the samples were only sent once every 5 seconds, the energy use of the radio is small. Figure 5.3 shows the lifetime of the system at different duty cycles for different battery capacities.

Since we have access to 80 Ah batteries that fit in our enclosure, we can estimate how long this system would last with larger capacity batteries. We can see even in the worse case, an 80 Ah battery would last longer than 1 year. If we were to sleep various components of the mote and if we were to do smart local data processing to transmit only deviations from a model, we could even further reduce power consumption and extend system lifetime. These optimizations are left for future work.

5.2 Occupancy Classification

In this section, we develop the process for occupancy classification in order to estimate occupancy from the ThermoSense node. Figure 5.4 shows an overview of this process. Information from the PIR and thermal sensors is used to update and maintain background levels of the thermal map. The current values along with the current background are used to create features vectors. These feature vectors are inputed into a classification model, which produces a "raw" occupancy estimate. A filter is then applied to the "raw" occupancy along with past occupancy estimates to produce the final occupancy estimate.

In the following sections, we develop and analyze each component of the occupancy classification process and examine the overall performance of the resulting occupancy estimates. We first examine the performance of the PIR sensor, and examine, in particular, the ability of the sensor to detect empty rooms. We next describe the process used to maintain the thermal background using the thermal array and PIR sensor. Then we describe how the background and current thermal map are used to create the feature vectors used for the occupancy classification model. We examine three different classification models; K-nearest-neighbors (KNN), linear regression (LR), and artificial neural networks (ANN). We then discuss the use of the filter on the "raw" occupancy output of the models. Finally,

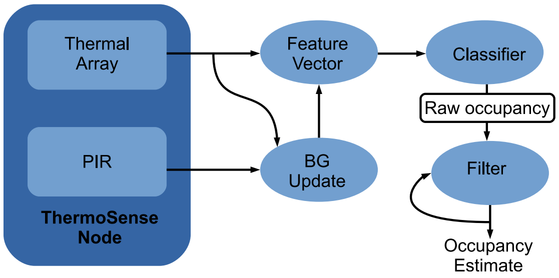


Figure 5.4: Occupancy Classification Process

we examine how the final output of the process performs by comparing the output to ground truth data collected over a 24-hour period of time from manually processed video feeds.

5.2.1 PIR Sensor Input

In our ThermoSense nodes, the PIR is used to detect if a room is currently occupied or unoccupied. The nature of the PIRs only allows motion detection, but people are not constantly moving when they are occupying a room. The PIR signal constantly fluctuates when motion is detected, and this requires a smoothing method to process it. Therefore, we smooth the raw PIR values over a period of 8 minutes using a moving average. This period was found by evaluating multiple different time windows and it is sufficient time to allow someone to remain inactive for a short time while still being able to detect that the room is being used. An 8 minute period was also found to return the least number of false positives. These false positives can occur when a person momentarily enters a room (e.g. a janitor) or positives that continue to be returned after a person has left a room (e.g. motion activity before leaving). This smoothing compared to ground truth can be seen in Figure 5.5.

Due to low number of false negatives, seen in Figure 5.6, the PIR can be used reliably as an unoccupied indicator. This can be used to override any predictions that would normally be made by our thermal array sensor. Once a room has been

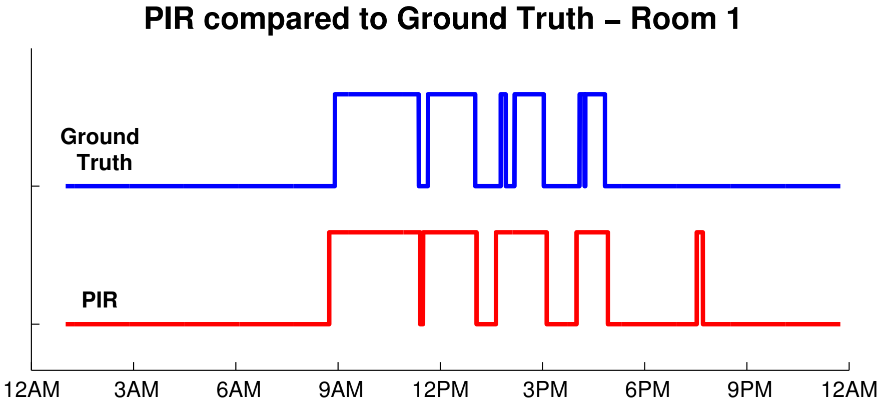


Figure 5.5: PIR compared to ground truth.

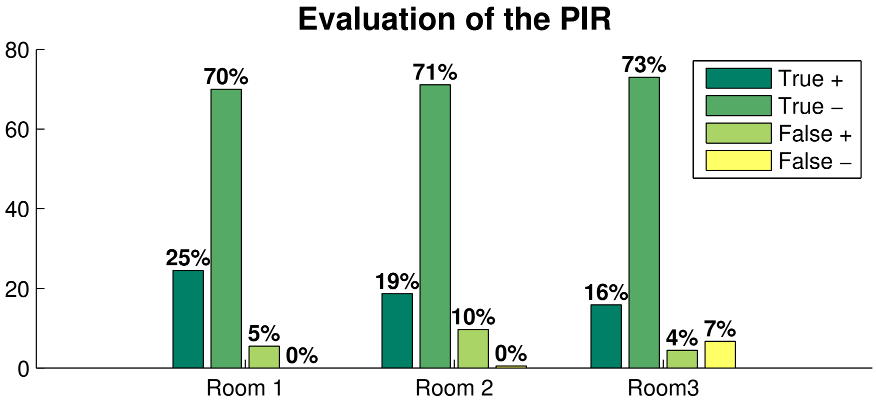


Figure 5.6: PIR evaluation of all three rooms

established as occupied by the PIR, additional models can be used to further evaluate the actual occupancy of a room.

5.2.2 Thermal Background

Since occupants are typically by far the warmest objects in a conditioned room, the thermal array can be used to detect occupants within a space. However, in order to distinguish between passive warm objects such as computers or refrigerators and humans, we maintain a thermal background map. If the PIR sensor indicates the room is empty, then this information can be used to determine the thermal background of the space. As the background can change over time, this background is continuously updated. In addition to maintaining the background, the standard deviation of each grid position is also saved; this standard deviation is used in the following section as a thresholding parameter for distinguishing significantly warm grid components. Algorithm 3 defines how our current background and standard deviation for each pixel for this background is maintained.

Before defining the feature vectors, we first discuss how we maintain our background temperatures. In order to define our features, background subtraction is used to differentiate an active pixel from a warm spot in a room. If the PIR has detected no movement for the 15 previous minutes, the background is updated using an exponential weighted moving average (EWMA) and the standard deviation is updated for each grid component. We found 15 minutes worked best, as it is unlikely an occupant remains motionless for 15 minutes. This threshold is also commonly used for PIR based lighting control [NLP98]. However, if an occupant occupies a space for a significant period, it is possible that the background changes during this period. To adjust the background while the space is occupied, we chose a few grid points with the lowest temperatures as our scaling components. The points with the lowest temperatures are most likely unoccupied and can be used to update the old background. We divide these scale points with the old background and average them to find a multiplier we can use to update our previous background. We then multiply the scale to the old background to find out a new background. The new background is smoothed into our old background by using an EWMA but with a significantly lower weight applied.

Algorithm 3 Background Update

```
$EWMA(a, x, y) \leftarrow$ EWMA with weight $a$.
$minTemp(f, n) \leftarrow$ indices of the $n$-lowest temperatures from frame, $f$
$frame \leftarrow$ current frame
$newBg \leftarrow$ returned updated background
$oldBg \leftarrow$ old background
$windowBG \leftarrow$ sliding window of the background

if PIR has been off for more than 15 minutes then
    $newBg = EWMA(0.1, oldBg, frame)$
    add $frame$ to $windowBG$
    threshold $\leftarrow$ 3 * std($windowBG$)
else
    $indices \leftarrow minTemp(frame, 5)$
    for each $index$ in $indices$ do
        $scalePx(index) \leftarrow oldBG(index)/frame(index)$
    end for
    $scale \leftarrow mean(scalePx)$
    $scaledBg \leftarrow scale * frame$
    $newBg = EWMA(0.01, oldBg, scaledBg)$
end if
```

5.2.3 Feature Vectors

We next define feature vectors that we will use as input for the classification models. While it is possible utilize all 64 values of the thermal array along with the PIR sensor, we found that this approach did not generalize well; the models would not work well when applied to different areas. Instead, we use the following three features for our models; the total number of significantly warm points, the number of grouped points that are warm, and the size of the largest group of warm points. The following subsections formally defines how we extract these features. We tried a significant number of other less successful feature vectors, e.g. 64 raw values, 64 thresholded-binary values, size of all connected components, different permutations of the feature vectors, etc. We try these different options but none of them work as well as the method chosen.

All three feature vectors are based on identifying the significantly warm points on the grid. As our first step, we first create a 8x8 binary matrix representing the significantly warm points from the thermal map. This is done by taking the difference between the background and the current thermal map and applying a standard deviation based threshold to this difference. We define current thermal values, background, and standard deviation as $M_{ij} = (m_{0,0}...m_{7,7})$, $B_{ij} = (b_{0,0}...b_{7,7})$, $S_{ij} = (s_{0,0}...s_{7,7})$, respectively. An active point is defined as being three standard deviations away from the background,

$$f(i,j) = \begin{cases} \text{True}, & \text{if } M_{i,j} - B_{i,j} > 3 * S_{i,j} \\ \text{False}, & \text{if } M_{i,j} - B_{i,j} < 3 * S_{i,j} \end{cases}$$

Feature 1: Total Active Points - For our first feature, we use the total number of active points. A larger total is correlated with higher number of occupants.

Feature 2: Number of Connected Components - Our second feature is based on connected components [HT73]. Connected components is a method of identifying groups of points within a matrix that are connected to each other. A point is considered connected if it is the same value as the point diagonal or directly adjacent. A component is a group of connected points. The number of the components, which are number of grouped warm points in our application, is correlated to the number of people occupying the area.

Feature 3: Size of Largest Component - The third feature is the size of the largest component with the grid. Multiple occupants standing close together will create one large component rather than several separate components, which would results in a lower occupancy count. The size of the largest component is positively correlated with occupancy and can be used to get a more accurate count.

5.2.4 K-Nearest Neighbors

The first model we consider is K-Nearest Neighbors. Let $X = (x_0 ... x_n)$ be the the components of the feature vector of the current frame. $Y = (y_0 ... y_n)$ is each individual feature components found inside the entire training set, $Z = (Y_0 ... Y_m)$. We use euclidean distance to calculate the distance between feature vectors,

$$d(X, Z_j) = \sqrt{\sum_{i=0}^{n} (x_i - Z_{ji})^2}$$

We then find the minimum k distances from the training set of Z and collect the distances as $D = (d_0 ... d_k)$ with it's corresponding occupancy labels as $L = (l_0 ... l_k)$. Weight is applied depending on the distance to each labels to get the final predicted occupancy, P.

$$P = \sum_{i=0}^{k} w_i l_i, \text{ where } w_i = 1 - d_i / (\sum_{j=0}^{k} d_j)$$

When calculating KNN, we find the 5 nearest neighbors and the distances associated with each. The predicted occupancy is averaged with the labels associated with the 5 closest values, with the largest weight given to values with the smallest distance.

	Value	p-value
β_A	0.141	2.44×10^{-187}
β_S	-0.051	1.14×10^{-30}
β	0.201	9.25×10^{-11}
F-statistic	3220	≈ 0
R^2	0.858	

Table 5.2: Parameters of linear model and fit metrics.

5.2.5 Linear Regression

The next model we examine is a linear regression model. We define a linear model

$$y = \beta_A A + \beta_S S + \beta$$

where y is the estimated occupancy (predicted variable), A is the number of active pixels and S is the size of the largest component (indicator variables). β_i is the corresponding coefficient for the indicator variable i and β is a constant. While the other models also included the number of components as a parameter, for the linear model, we chose A and S by testing permutations of the feature vectors that minimized root mean squared error (RMSE) and had significant coefficients ($p < 0.05$). We also found that the model fit best when estimating positive occupancy. Thus, we train the linear model with data for the 1, 2, and 3 person cases and rely on the PIR sensor to determine if a space is empty.

Table 5.2 shows the fitted model parameters along with p values of each parameter and the F-test of the overall model. Using a $p < 0.05$ threshold, we see the p-values for each of the indicator variables is significant. We see also find that $p \approx 0$ from the F-test and $R^2 = 0.858$, indicating a good fit. Figure 5.7, shows the distribution of the residuals. The normal distribution verifies that the independent error assumption has not been violated.

5.2.6 Artificial Neural Network

The final model we consider is a forward feed ANN [Mit97] using a single hidden layer of 5 perceptrons. We use a sigmoid for the hidden layer transfer function and a linear transfer function for the output layer. The same data for the 1, 2, and 3 person case was used for the ANN model. Again we use the PIR value to determine the 0 person case. We used 70% of the data for training, 15% for testing, and 15% for validation. We found $R^2 = 0.893$ and $R^2 = 0.906$ when compared with the testing set and the entire dataset respectively, suggesting the model has a strong fit.

5.2.7 Filter

The last step of the occupancy classification process is the filtering of the raw occupancy estimate. If we examined the error of the models, we found that the errors are independent and normally distributed (see Figure 5.7). Thus, we employ a 4 minute moving average filter in order to reduce error of the raw occupancy estimate produced by the classification model. We found that the 4 minute window minimized our error and was found through trial and error. Figure 5.8 compares the raw output of the model with the filtered output. From the figure, we can see that the filter is effective at removing independent errors.

5.2.8 ThermoSense Performance

In this section, we examine the performance of the process with respect to the three models. In order to test the performance, we gathered 24 hours of ground truth data. This data was gathered by deploying a camera in a public hall way and manually counting the number of people entering and leaving the zone through out the day. For our analysis, we examined a single zone comprised of 3 separate offices. For our performance metrics, we use RMSE and normalized root mean

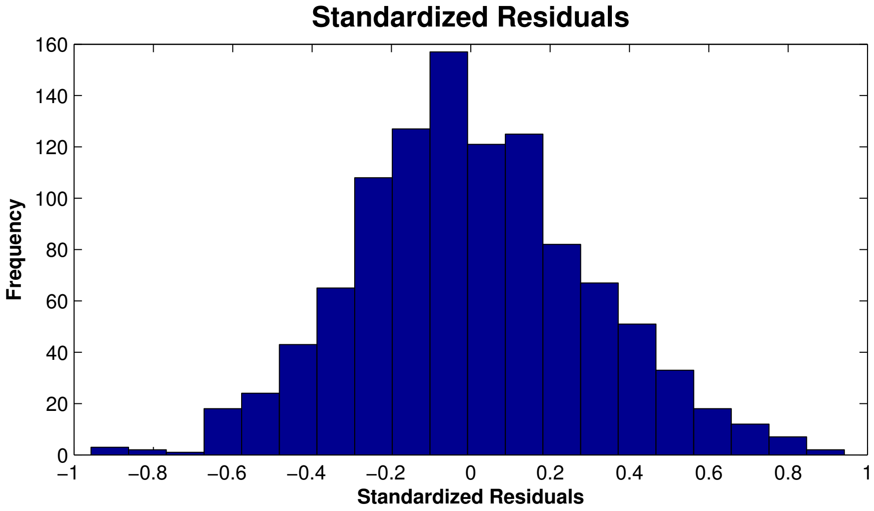


Figure 5.7: Plot of residual distribution

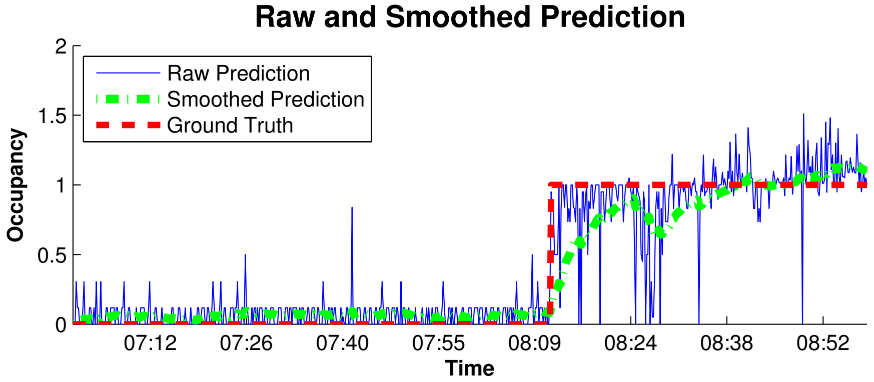


Figure 5.8: Model raw and filtered outputs.

squared error (NRMSE).

Table 5.3 summarizes the performance results. Overall KNN, performed the best. KNN had a RMSE of 0.346 people (NRMSE of 11.5%). LR and ANN had slightly higher RMSE values of 0.409 people and 0.385 people (NRMSE of 13.6% and 12.8%) respectively. Figure 5.9 shows the performance of the models as a function of the amount of training data available. In particular, KNN is able to perform fairly well relative to the amount of data available. With only 100 training samples, KNN has a NRMSE value of 25%. Both LR and ANN have values 35% and 34% respectively when trained with the same 100 samples. LR and ANN have very similar performance between 100-900 samples. After 900 samples, we see that the ANN starts to have slightly better performance than LR. Though KNN maintained lower NRMSE overall, the difference among the different models became small as the available data increased; the maximum difference among models is only 2.1%.

5.2.9 Model Discussion

While each model had similar results with KNN having the best RMSE, there are other issues to be considered. KNN required the least amount of data in order to achieve a low NRMSE. However, Figure 5.9 shows that as more training data is available, the other models may well match or outperform KNN. Another consideration is that as the number of training data increases, the run-time performance of KNN will decrease. While there is a possibility of running KNN on a mote, it requires storing the entire training set and iterating through the set for each new sensed sample. Thus, KNN is more suited to situations with little data exists and KNN is processed at the base-station rather than the node. Though the linear model had the lowest accuracy, there are other advantages to this approach. In particular, it can be run very efficiently on a mote; in our case, only the values of the coefficients $(\beta, \beta_A, \beta_S)$ need to be stored on the mote and new estimates are

	KNN	LR	ANN
RMSE	0.346	0.409	0.385
NRMSE	11.5%	13.6%	12.8%

Table 5.3: Evaluation of the models used.

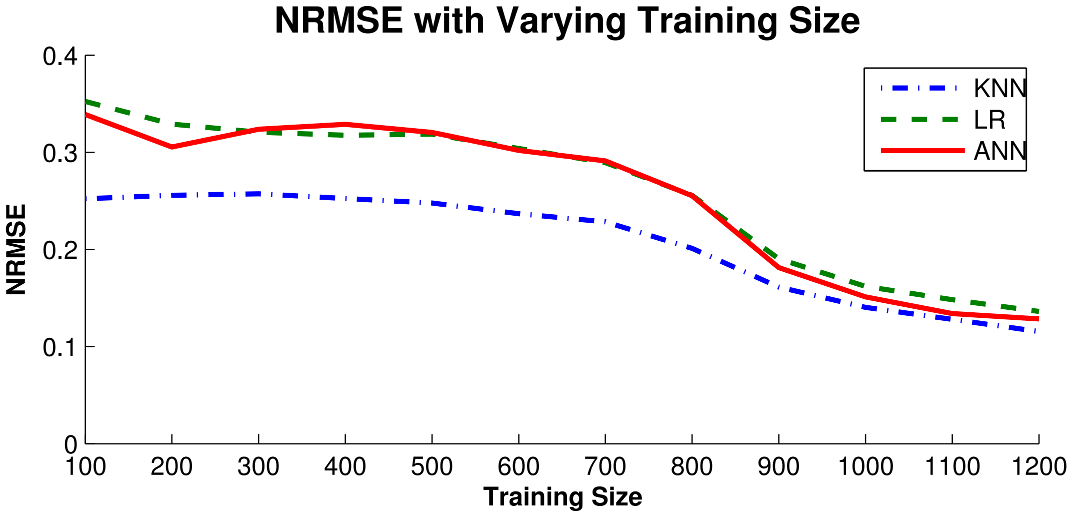


Figure 5.9: NRMSE of a ThermoSense node in a zone.

simple to calculate in real-time. A similar argument can be made for ANN; only the input and weights need to be stored on the mote and estimates are trivial to calculate in real-time. ANN can require potentially more time to train, but as this is a one time cost, ANN may be preferable over LR since ANN can handle non-linearities.

5.3 Summary

In this chapter, we developed ThermoSense, an occupancy monitoring system that utilizes thermal based sensing and PIR sensors. We developed a novel low-power multi-sensor node for measuring occupancy using a thermal sensor array combined with a PIR sensor. The thermal sensor array is able to measure temperatures in an 8x8 grid pattern within an 2.5mx2.5m area when deployed at ceiling heights of 3m.

We showed it is possible to use these temperature readings in order to determine how many people are within the space using a novel occupancy classification process. Unlike a CO_2 sensor, the thermal array can measure occupancy in near real-time. The thermal array is also not sensitive to optical issues associated with the strategy discussed in the last chapter, such as lighting or background changes. By adding a PIR sensor, we increase the accuracy of detecting empty spaces and the overall accuracy of the platform. The PIR sensor is also used to reduce the node's power consumption by triggering the mote and thermal array only when someone is present. We tested this new platform with a 17-node deployment covering 10 building conditioning areas totaling 2,100 sq. ft. for a period of three weeks and showed that ThermoSense is able to detect occupancy with a RMSE of *only* $\approx$0.35 persons. Though we found KNN to work the best, both ANN and linear regression worked sufficiently well for the purpose of estimating occupancy for HVAC systems.

CHAPTER 6

OBSERVE: Occupancy-Based System for Efficient Reduction of hVac Energy

A major goal of this dissertation is to condition based on actual use rather than assumed use. In order to accomplish this, we need knowledge of occupancy. In Chapters 4 and 5 we developed two different wireless sensor networks for measuring occupancy in near real-time; this will allow us to provide the appropriate amount of ventilation to the spaces. Next, we need to know when people occupy the space so heating and cooling can be actuated beforehand. This needs to be done predictively since time is required for rooms to reach target temperatures. In Chapter 3, we developed several methods of predicting occupancy. With occupancy modeling and occupancy measurement, we are now able to develop control algorithms that can predictively actuate heating and cooling beforehand and ventilate based on actual occupancy.

In this chapter, we combine the systems and modeling developed in the previous chapters to create a conditioning strategy. In this chapter we contribute the following:

- We demonstrate how models developed from sensor network data can be integrated with an HVAC control strategy to achieve significant energy savings.
- We show that we can still satisfy ASHRAE conditioning standards while saving significant energy. In certain cases, we show that our predictive

strategies meet user demand better than typical baseline strategies.

- We examine a PIR based WSN solution and show that binary measurement of occupancy is not always sufficient for HVAC control. Under some circumstances, PIR based HVAC control performs worse than current strategies; in order to achieve maximum energy savings within a building, *accurate real time levels of occupancy must be incorporated into HVAC strategies.*

6.1 HVAC Conditioning Strategy

We start by examining the basic criteria required for usage based occupancy conditioning. In Chapter 3, we explored several different occupancy strategies. The one that best represented occupant behavior based on the metrics examined was the Blended Markov Chain (BMC). Thus, for our control strategies, we will use BMC to define a predictive control algorithm for temperature. We start by defining the thermal and ventilation criteria that our HVAC strategies must meet to have an ASHRAE compliant building. We then define the occupancy based temperature control strategy.

6.1.1 Conditioning Criteria

6.1.1.1 Temperature

Thermal comfort is a complex measurement that depends on many aspects such as temperature, humidity, air velocity, occupants clothing and activity [PHK10, PHK07]. The most common comfort measurement is Fanger's Predicted Mean Vote PMV as standardized in ISO 7730 [OP02]. Fanger's model is not an undisputed thermal comfort measurement [OP02, PHK07, SBW10], but provides good generalized results in many cases. Fanger computes the thermal comfort PMV value between -3.5 and 3.5. Rounding the PMV results in the comfort classes hot (PMV > 2.5), warm ($1.5 \leq$ PMV < 2.5), slightly warm ($0.5 \leq$ PMV < 1.5), neutral ($-0.5 \leq$ PMV < 0.5), slightly cool ($-1.5 \leq$ PMV < -0.5), cool ($-2.5 \leq$ PMV < -1.5), and cold (PMV < -2.5). Fanger's PMV depends on air temperature [OP02], radiant temperature, humidity, air velocity, occupants clothing and activity. The model is non-linear and based on large scale studies in climate chambers.

Measuring the PMV based on this model is very complex due to the many influences. The activity and clothing level of occupants is simplified by assuming defaults such as office work and clothing level is correlated to the outside temperature [4]. However, this still requires an air temperature, radiant temperature, humidity and air velocity sensor in each room. Air temperature and simple humidity sensors could be added to our wireless sensor node, but radiant temperature and air velocity sensors are complex and expensive, such that large scale installations are not affordable. Given these complexities, our work focuses on optimally controlling temperature rather than attempting to control PMV. From a control perspective, our goal is to meet a target temperature defined by a comfort metric, which may not necessarily be Fanger PMV. Different comfort metrics will establish different temperature set-points. It is thus more important to meet specific target temperatures than to meet a specific comfort metric. As metrics are developed and better environment sensing is available, we want our system meet the temperature goals dictated by the new comfort criteria.

6.1.1.2 Ventilation

ASHRAE Standard 62.1 [ASH07b] uses the following to calculate outdoor air ventilation rates:

$$V_{bz} = R_p P_z + R_a A_z \tag{6.1}$$

where z denotes the zone, V_{bz} is the ventilation rate, R_p is the minimum CFM/person, P_z is the number of people, R_a is the minimum CFM/ft^2, and A_z is the floor area. The P_z by determining A_z. The other constants are determined by ASHRAE Standard 62.1.

6.2 OBSERVE Temperature Control Strategy

This section defines the Occupancy-Based System for Efficient Reduction of hVac Energy (OBSERVE) predictive control algorithm, which uses a BMC to predictively condition room temperature. The function $BMC(OccState, predLen)$ return a predicted schedule of occupancy $occPred$, where $occPred_t$ is a probability a space being occupied at time $t = 1 \ldots predLen$ and $OccState$ is the occupancy state at time $t = 0$. This conditioning strategy is also applicable for binary occupancy data. The BMC can be easily adapted to binary data by representing the MC state at each time as a vector where each component is a binary indication of occupancy (instead of actual occupancy). Algorithm 4 defines the OBSERVE temperature control strategy.

This strategy conditions for two different types of occupancy. It first checks short windows of time for occupancy to ensures that rooms, such as copy rooms, that are not frequented often but have occupied durations of several minutes are conditioned. The second longer window ensures rooms that are frequented constantly but may have short stay durations are still conditioned. Hallways are an example of such an area. Since no standards currently define what constitutes a significant duration of occupancy, we chose values that seem reasonable.

Algorithm 4 OBSERVE temperature control algorithm.

```
$CondTemp_{i,j} \leftarrow$ Condition temperature from time $i$ to $j$

$CurrHour \leftarrow$ Current hour

$T_{TG} \leftarrow$ Temperature such that $PMV = 0$

$T_{ASH} \leftarrow$ Temperature such that $-0.5 < PMV < 0.5$

$pThresh \leftarrow$ Probability threshold of occupancy


for Every $n$ minutes do
    $CurrOcc \leftarrow$ Current occupancy state
    $occPred \leftarrow BMC(CurrOcc, predLen)$

    for Each room $r$ and point of time $t$ in $occPred$ do

        $occupied \leftarrow$ All periods $occPred_{t \rightarrow t+60} > pThresh$

        if $occupied > 5$ minutes of next 15 minutes then
            $CondTemp_{t-60,t+15} = T_{TG}$
        else if $occupied > 20$ minutes of next 60 minutes then
            $CondTemp_{t-60,t+60} = T_{TG}$
        else if $5 \leq CurrHour \leq 24$ then
            $CondTemp_{i,i} = T_{ASH}$
        end if

    end for
end for
```

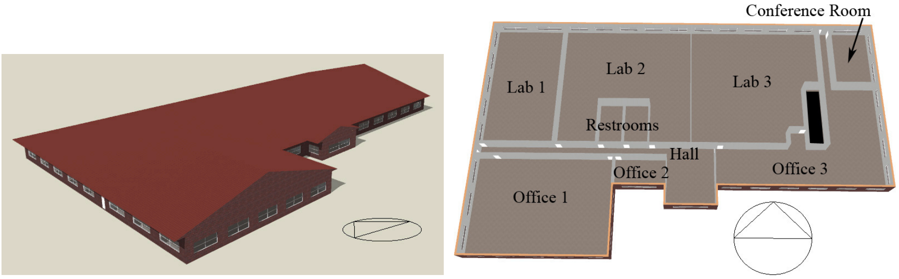
6.3 Evaluation

Four different strategies are considered. The first is coefficient based baseline strategy (ASHRAE baseline) that employs a typical HVAC control strategy assuming maximum occupancy for ventilation and conditions all rooms from 7:00 - 22:00. The second strategy (Reactive Binary) is a binary based reactive control of temperature and ventilation. Rooms are conditioned to a target temperature when occupied and are not predictively preconditioned. Rooms that are unoccupied are still conditioned to the max/min temperatures allowed by ASHRAE. Since precise room occupancies are unknown, rooms are ventilated according to the estimated maximum occupancy when occupied. The maximum occupancy is determined using ASHRAE Standard 62.1 based on area and room purpose. The third strategy (OBSERVE Occupancy) uses the OBSERVE algorithm described in Section 6.2. Ventilation is based on the observed room occupancy using equation 6.1 during warm months (Apr - Oct). During the colder months ventilation rates are increased. This increased ventilation is an optimization for terminal reheat HVAC system and is discussed in Section 6.4.2.1. The last strategy (OBSERVE Occupancy) uses the OBSERVE algorithm but uses binary data to train the BMC model. This strategy makes binary predictions of occupancy and ventilates with the same maximum occupancy assumption of the reactive control strategy.

6.4 Performance Results

6.4.1 Building Energy Simulator

EnergyPlus is one of the premiere tools for modeling the energy of buildings. It takes into account factors such as weather, HVAC design, and construction materials. In this section we define the main parameters used for an EnergyPlus



EnergyPlus Building Parameters

HVAC	Single Duct AHU, VAV with terminal reheat Gas heating and cooling 9 zones
Temperature Setbacks	Heating: $T_{tg} = 75^o$F, $T_{ASH} = 70^o$F Cooling: $T_{tg} = 78^o$F, $T_{ASH} = 82^o$F
Areas	Total: 30,130 ft^2
Materials	Concrete exterior walls Metal interior walls 30% double glazed windows

Figure 6.1: Main building parameters used.

model to test our HVAC strategies.

6.4.1.1 Building Parameters

EnergyPlus simulations are run for three different locations; Fresno CA, Miami FL, and Chicago IL. The EnergyPlus model replicates the geometry shown in Figure 6.4.1 and is constructed to ASHRAE standards. Figure 6.1 summarizes the main model parameters. The HVAC is a terminal reheat system that uses a single air handler unit (AHU) with variable air volume vents (VAV). The sizing of the AHU is done according to ASHRAE Standard 62.1 [ASH07b]. This type of HVAC system cools air at the AHU level and heats primarily at the VAV level. This is common for many office buildings [BK09]. This type of system is often

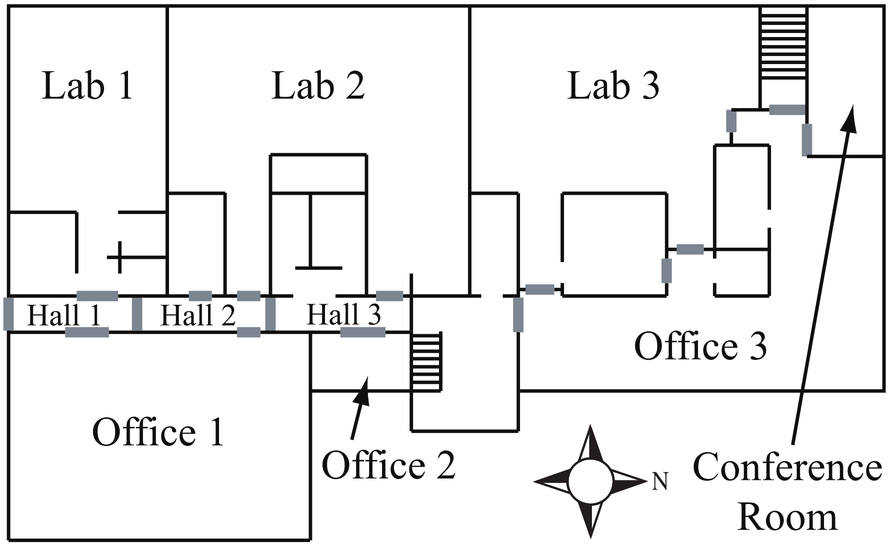


Figure 6.2: The left shows the ten areas data was collected for and the 18 boundaries (gray lines) defining the areas.

used since it is able to heat and cool areas while still sharing the same AHU. We save the evaluation of other types of HVAC systems for future work. The set-points correspond the temperatures that predict $-0.5 \leq$PMV≤ 0.5 assuming 40% humidity, 1 m/s airflow, and clothing coefficients of 1.0 and 0.5 for winter and summer respectively. Fresno, Miami, and Chicago have 40%, 52%, and 32% average humidities respectively. The HVAC system controls humidity to be between 35% and 45% using a cool-reheat heating coil.

6.4.1.2 Simulated Occupancy Schedules

EnergyPlus models typically use static occupancy schedules specified in sources such as ASHRAE 90.1 and DOE-2 [Doe12] and are often based on survey data. For our model, however, we will use the BMC to generate occupancy schedules. A BMC trained with 5 days of ground truth data is used to generate 23 days of simulated "ground truth" occupancy data for the building model. These simulations will serve as our occupancy schedules for the weekdays of the month. The

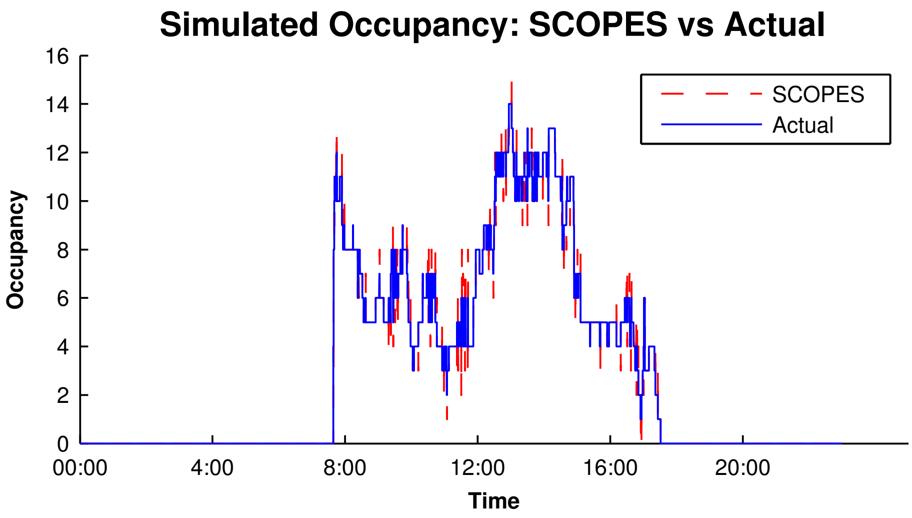


Figure 6.3: Simulated SCOPES data.

building is assumed to be empty on weekends. Static occupancy schedules are used for the restrooms since we have no data for privacy reasons.

Our control strategy assumes a system similar to SCOPES for occupancy monitoring. For 20% of the time, the system inverts the direction or detects a false positive or negative transition. We simulate SCOPES system error by artificially introducing errors into the simulated ground truth data. Because directional errors occur with roughly the same frequency, even with 80% accuracy, the observed system data is close to ground truth. Since errors may produce states that do not exist in the BMC, the closest distance metric defined earlier is used to find the closest state for the OBSERVE algorithm. Figure 6.3 compares the simulated scopes with the simulated ground truth.

6.4.2 Energy Savings

We evaluate the energy savings possible using the defined HVAC strategies for each location. These results take into account the fan, pump, heating (gas), and cooling (gas) energy consumption of the building. Figure 6.4 shows the monthly

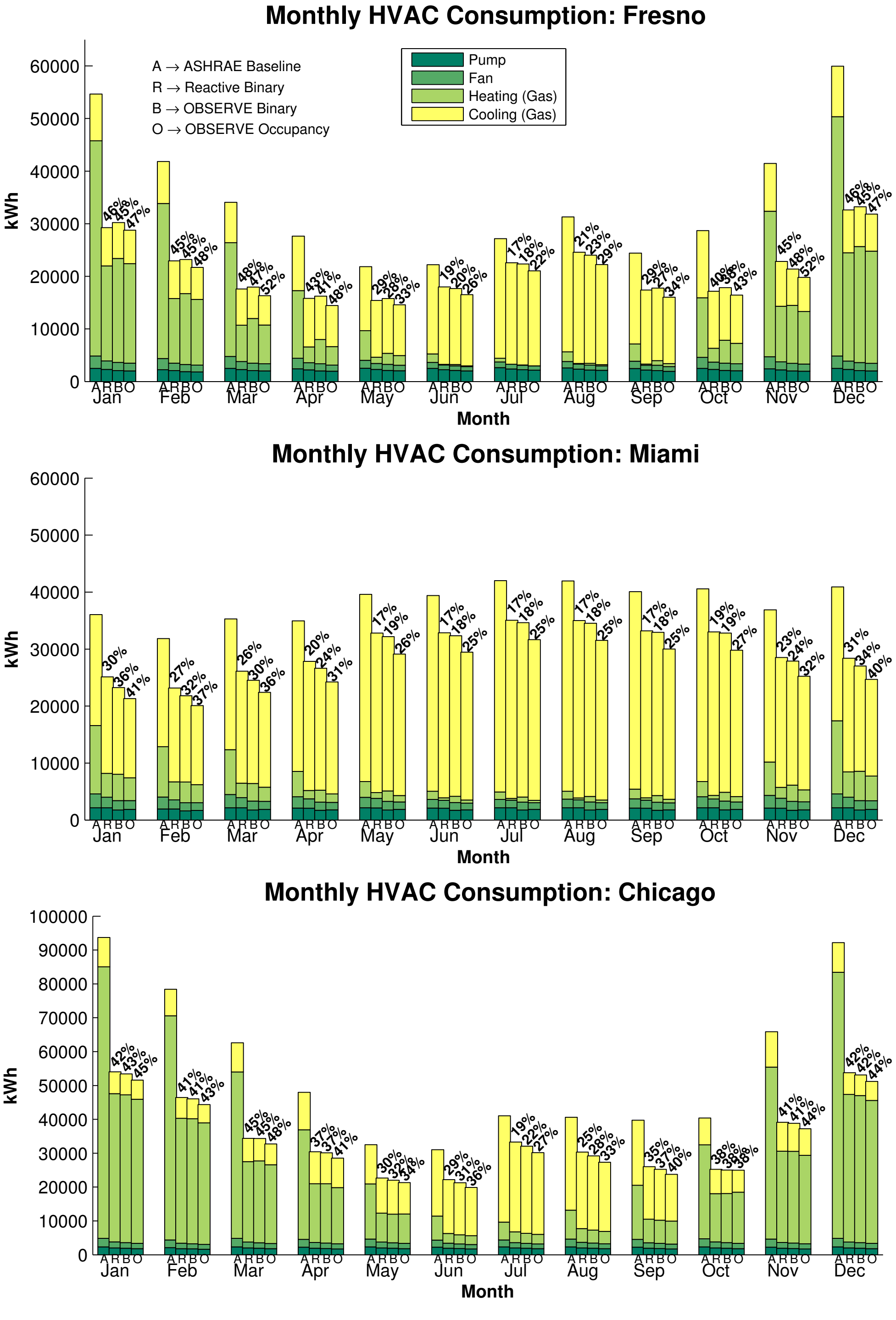


Figure 6.4: The breakdown of the energy consumption for each month and strategy for three different locations.

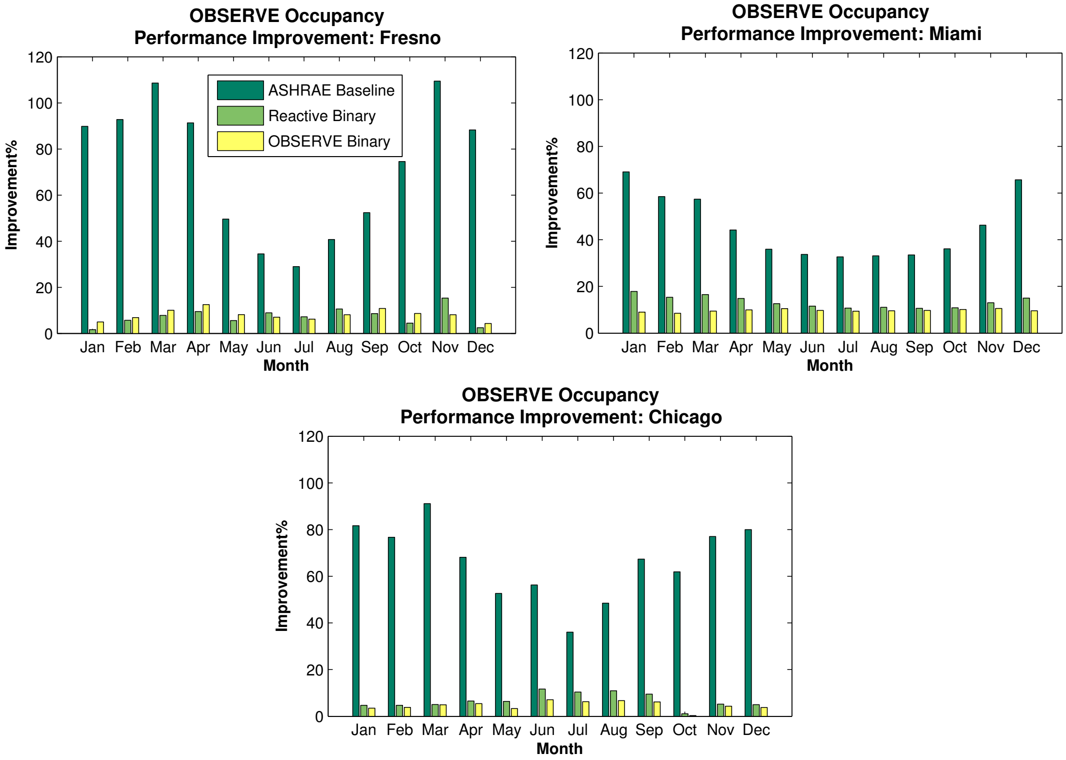


Figure 6.5: Improvement of energy consumption over baseline.

breakdown of the energy consumption for each location. Significant energy savings are achievable over a standard commonly used baseline; we see up to 112% improvement (Figure 6.5) in some cases. We see that heating and cooling account for most of the energy usage and that heating requires more energy than cooling. OBSERVE Occupancy out-performs binary strategies. The strategies have the most substantial savings during the colder months. This is primarily because heating costs are greater than cooling costs.

During the coldest months in Fresno, the strategies are 45%-47% more efficient than baseline; in particular OBSERVE Occupancy shows 88% to 112% improvement over baseline. For Fresno, OBSERVE Occupancy, OBSERVE Binary, and Reactive Binary show annual savings of 42.3%, 38.3%, and 37.9% respectively. Similar results are found for the Chicago location. The difference among strategies increases during the warmer summer months. For summer months in Fresno, OBSERVE Occupancy shows 9%-11% improvement over Reactive Binary and 7%-10% improvement over OBSERVE binary. For summer months in Chicago, OBSERVE Occupancy shows 10%-12% improvement over Reactive Binary, and 6%-7% improvement over OBSERVE Binary. This suggests OBSERVE Occupancy is more efficient for warm weather areas. When we examine the results from the warmer Miami location, OBSERVE Occupancy shows 11%- 18% improvement over both binary strategies throughout the year. OBSERVE Occupancy again shows significant improvement over baseline (33%-69%). We see that OBSERVE Occupancy is 26%-41% more efficient than baseline whereas Reactive Binary is 17%-30% more efficient. OBSERVE Occupancy, OBSERVE Binary, and Reactive Binary have annual savings of 30.4%, 23.7%, and 21.4% respectively for Miami.

6.4.2.1 Discussion

We see that all three strategies are significantly more efficient than baseline and that there are noticeable differences among the strategies. The main source of these differences is related to ventilation and the terminal reheat system used for the building. OBSERVE Occupancy is able to adjust the ventilation rate according to estimated occupancy whereas OBSERVE Binary and Binary Reactive use the maximum ventilation rate during periods of occupancy. Typically, increasing the ventilation rate introduces additional outside air into the air handler loop and increases the energy consumption since the outside air needs to be conditioned; re-circulating air is more efficient since it takes less energy to maintain pre-conditioned air. However, there are other ventilation efficiency factors that need to be considered with a terminal reheat system. A terminal reheat system partially preheats outside air before entering the AHU then reheats air at the room VAV. Under certain conditions, it is more efficient to increase ventilation. Increasing ventilation means losing some energy since more outside air needs to be conditioned. However, increased ventilation distributes the heating load across all the VAV's and increases the amount of VAV pre-heated air into the loop. During the colder months, occupant load helps to increase the efficiency. The same is not true for cooling. In this case, additional occupants decrease efficiency.

6.4.3 Conditioning Effectiveness

While substantial energy savings are possible, we must verify the building is ASHRAE compliant. For this analysis we focus on the Fresno EnergyPlus simulations.

6.4.3.1 Temperature Effectiveness

We examine the temperature of the rooms during all times of occupancy. In particular, we are interested in areas such as the Conference Room, which are not occupied the majority of the day. In order to be ASHRAE [ASH07a] compliant, we must maintain the set-point temperatures to ensure $-0.5 \leq PMV \leq 0.5$. As mentioned before, we examine temperature directly rather than PMV; our goal is to meet the temperature set by any metric of thermal comfort. For this analysis, we examine the root mean square error (RMSE) of the room temperatures. We also examine building orientation to observe the influence of solar gain. The results are shown in Figure 6.6.

All four strategies perform similarly for Office 1. Reactive Binary has the highest RMSE (1.3^o) for Nov-Mar. Both OBSERVE strategies have similar results throughout the year. Baseline performed best during the coldest months.

OBSERVE Occupancy performs the best out of the four strategies for Office 3 with a RMSE of 0.1-0.3 F^o. OBSERVE Binary has a slightly higher RMSE of 0.1-0.5 F^o. The Reactive Binary strategy is highest with a RMSE of 0.9-2.8 F^o. For Dec-Jan Reactive Binary has a RMSE of 2.5 F^o as compared with the RMSE

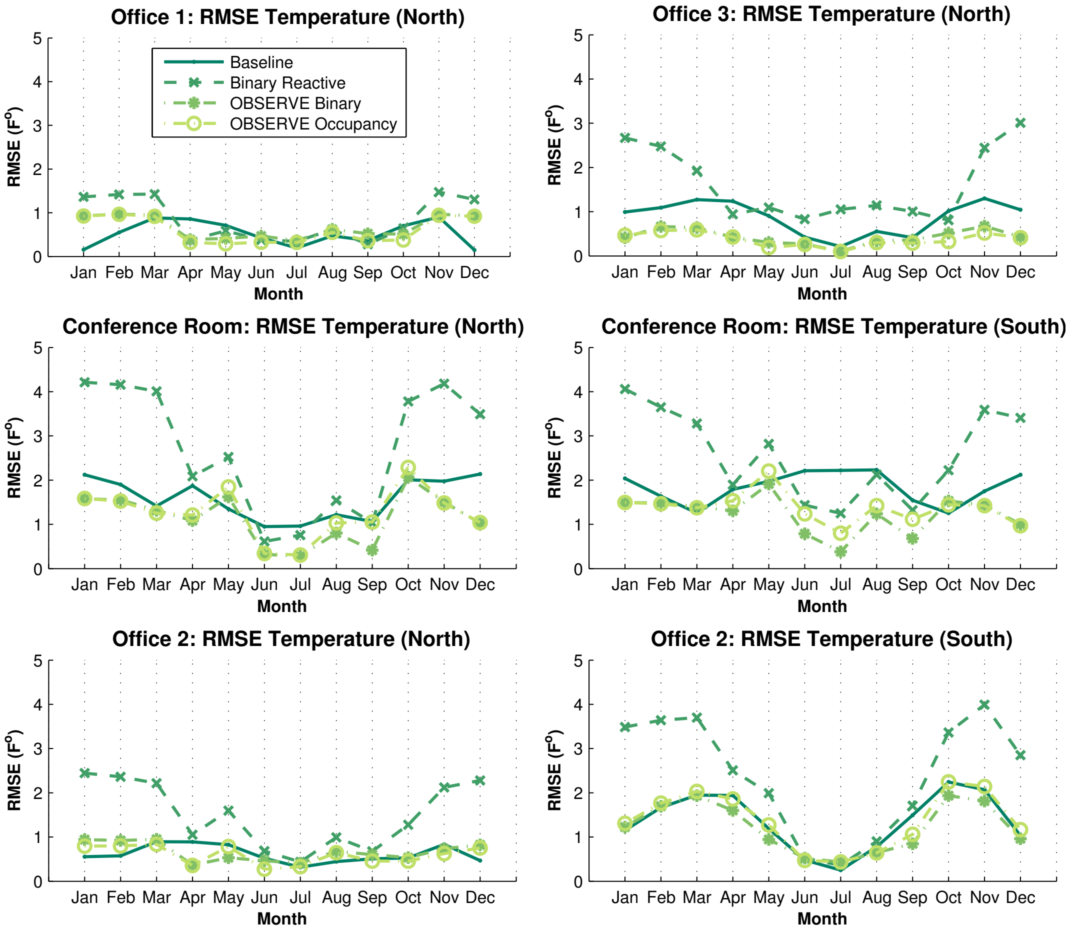


Figure 6.6: Monthly RMSE temperature for each strategy for Fresno.

of 0.1 F^o and 1.0 F^o for OBSERVE Occupancy and baseline respectively.

Two different building orientations are considered for Conference Room and Office 2 to show the effect of solar gain. North is defined in Figure 6.4.1. South refers to the building turned 180^o. Reactive Binary has higher RMSE for the south orientation (0.4-3.6 F^o) than the north orientation (0.2-2.4 F^o). Both OBSERVE strategies have lower RMSE than Reactive Binary for both orientations. Generally, both OBSERVE strategies have slightly lower RMSE than baseline for the north orientation. Both OBSERVE strategies show higher RMSE for the south orientation than the north orientation; this is caused by solar gain.

For the Conference Room, Reactive Binary has higher RMSE during the colder months (4.0-4.2 F^o) for both orientations. During the summer, Reactive Binary has higher RMSE for the south orientation. For the north orientation, the baseline does better than OBSERVE strategies for winter and spring. During the summer, however, both OBSERVE strategies have lower RMSE than baseline. Comparing OBSERVE strategies, OBSERVE Binary has lower RMSE than OBSERVE Occupancy for the summer south orientation (0.3-1.1 F^o vs 0.8-1.2 F^o).

6.4.3.2 Discussion

Reactive Binary tends to have higher RMSE than the other strategies because of the slow ramp up to target temperature. Once an occupant enters a room, it takes time to reach the target temperature. For areas occupied most of the time such as Office 1, Reactive Binary performance improves but does not match quality of service of predictive strategies since there is still a temperature reaction delay at the beginning of the day when the first occupant enters. This delay becomes more critical for areas sporadically occupied such as Office 2 and the Conference Room. A room occupied once or twice a day, such as the Conference Room has insufficient time to reach the target temperature and is only at the correct temperature if the outside conditions happens to match the target temperature (spring, fall). RMSE increases if the room receives afternoon sun as is shown for the southern orientation of the Conference Room and Office 2. Rooms such as Office 2 that are constantly visited have slightly lower RMSE since the thermal momentum creates an additive effect that helps maintain temperature. These results show that reactive strategies based on a PIR WSN *do not work in practice and cannot reach the target temperature* required by the users for all the different scenarios tested.

In some cases, OBSERVE strategies perform better than the baseline since the baseline uses a static HVAC schedule. This is seen for Office 3, which is occupied by professors and graduate students working at odd hours. The static

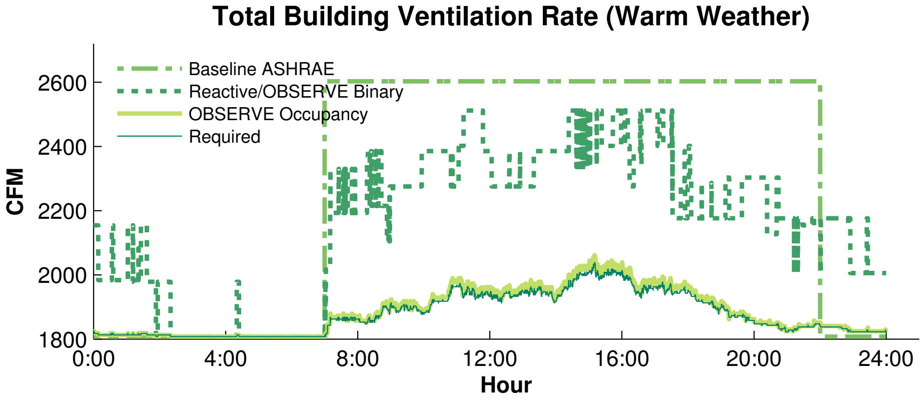


Figure 6.7: Total building ventilation rates for each strategy for one particular summer day.

baseline schedule stops conditioning during normal working hours whereas OBSERVE anticipates after hours usage. Even Reactive Binary out-performs the baseline during mild months (spring and fall) when rooms are occupied off-hours and the outside and target temperatures are close.

Both OBSERVE strategies perform similarly. However, there are instances in warm weather where OBSERVE Binary performs slightly better. This is related to over ventilation. Assuming maximum occupancy forces the VAV vent to open fully when conditioning a room and less time is required to cool the room. The same is not true for heating since air from the loop is only partially preheated.

6.4.3.3 Ventilation Effectiveness

The minimum ventilation rate is established by Equation 6.1. Since the occupancy detection system can undercount occupants, we add an additional 10% to occupancy estimates. Figure 6.7 shows the ventilation rate for a summer day. The baseline strategy assumes a maximum ventilation rate. The binary strategies use a maximum ventilation rate only when an area is occupied. OBSERVE Occupancy uses a modified schedule. It uses maximum ventilation during cold months

(Oct-Apr for Fresno/Chicago) and optimal ventilation during the warm months (year-round for Miami). From the summer day ventilation, we can see that the baseline greatly over-ventilates the building. While binary based strategies are an improvement, we can see that binary ventilation rate far exceeds the required ventilation rate. It has a RMSE of 335.0 CFM (Normalized RMSE of 146%). OBSERVE however, remains close to the required ventilation rate and has a RMSE of 9.5 CFM (NRMSE of 4.1%). OBSERVE is usually above the required ventilation because of the 10% safety margin for under-counting. OBSERVE fell below the required ventilation rate only 0.0004% of the time. From Oct-Apr, both binary and OBSERVE share the same ventilation rates and have a NRMSE of 146%.

6.5 Conclusions

In this chapter, we have shown how a model can be integrated with a WSN for demand control based conditioning strategies. We propose the OBSERVE predictive demand control strategy and test the energy savings and conditioning performance. We learned several lessons from our results. First, in order to achieve energy savings, *real time occupancy data is critical.* This can be seen by the average 42% annual energy savings compared to the current state of the art baseline strategy. Second, *predictive* strategies show better energy savings performance and even *significantly* better quality of service conditioning than reactive strategies. Finally, in order to achieve *maximum* energy savings, *actual level of occupancy* is required in order to optimize ventilation levels.

In order to develop OBSERVE, we used simulations based on real-world data. The next step is thus to test OBSERVE in a live deployment. In the next two chapters, we test the OBSERVE control algorithm using the occupancy monitoring solutions from Chapters 4 and 5.

CHAPTER 7

POEM: Power-efficient Occupancy-based Energy Management System

Using the control strategy from the previous chapter, we next test the control strategy in an actual deployment. In this chapter, we develop the **P**ower-efficient **O**ccupancy-based **E**nergy **M**anagement System (POEM), which shows how a control strategy can be implemented using the BMS programmatic interface to optimally condition rooms based on current and predictive occupancy. This system uses a control strategy similar to OBSERVE (Chapter 6) that has been adapted for the optical occupancy measurement system from Chapter 4. We evaluate the energy saving and the conditioning effectiveness with respect to temperature and ventilation and show that the cost of the system can be amortized in approximately 3 months if applied to the entire building. In this chapter we contribute the following:

- Our most significant contribution is the design, implementation, deployment and evaluation of the POEM system, which is a full close loop system that conditions rooms based on occupancy. This system combines observed occupancy, predicted occupancy, and data fusion for tracking occupancy using a particle filter in order to optimally condition rooms.
- We tested our system over a period of four weeks and show that significant energy savings (26%) are possible while still maintaining conditioning effectiveness. Using a calibrated EnergyPlus simulation, we show that this

system saves 30% energy annually over standard strategies.

- We perform Return On Investment (ROI) analysis, showing the sensitivity of different factors and concluding that the cost of the system could be amortized in approximately 6 to 10 months.

7.1 Overview

POEM is a complete system that controls the temperature and ventilation of a building. This system is comprised of several components. Figure 7.1 summarizes the system architecture and shows all components of the POEM system. OPTNet, a wireless camera sensor network, and BONet, a PIR sensor network, provide occupancy estimates (Chapter 4). A particle filter then fuses the sensing data from OPTNet and BONet with the output of an occupancy transition model in order to better estimate the current occupancy in each room. The current occupancy estimation from the particle filter, the ramp-up time, and the predicted occupancy from the transition model 1 hour into the future are combined in a control schedule to optimally pre-condition the spaces to reach the target temperature. Thus, we can optimally schedule the HVAC system to match the likely arrival of occupants. Once the process is complete, the conditioning affects occupants completing the feedback loop. The next sections provide a detail description and evaluation of each POEM component.

7.2 Actuation Scheduler Interface

We start be examining the programmatic interface that is available for the building automation system (BMS). The building being controlled utilizes both Automated Logic and Phoenix BMS systems [Log, pho]. The system is maintained through a web server/interface called WebCtrl [Log], which can issue BACnet commands

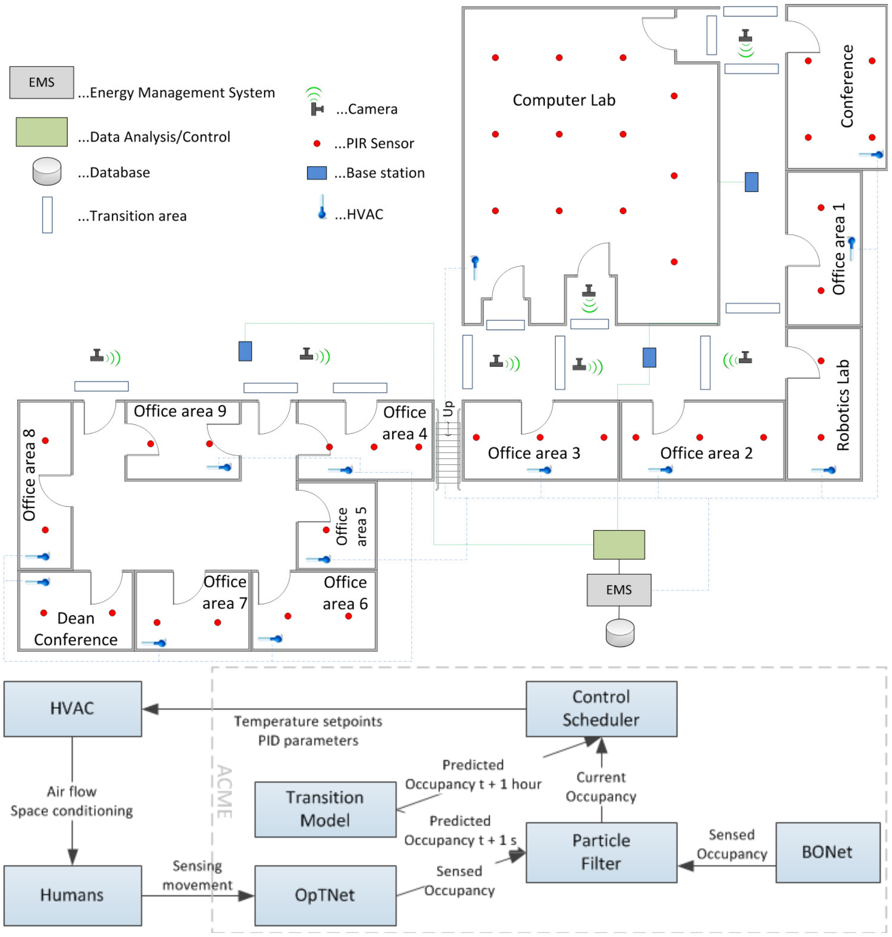


Figure 7.1: Architecture of the POEM system.

to the various HVAC components including the variable air volume (VAV) unit. Since the sequence of operations are stored on this server, the building energy manager indicated that it would be preferable to keep the logic of the system intact and change only the temperature/ventilation set-points within the server logic blocks rather than bypass the server and issue BACNet commands directly. Since the server could accept set-point changes via SOAP, we achieved control using python scripts.

7.2.1 Actuation Algorithm

As previously mentioned, near real-time occupancy is useful for accurate ventilation. However, occupancy prediction is necessary since time is required to reach the target temperature of a room. In order to determine the minimum time to reach a target temperature, we determined the maximum amount of time it was required to reach set-point given the setbacks. Again, occupancy is predicted using the blended Markov Chain approach used in Chapter 3. The main difference in the application of the BMC model for actuation/control is that occupancy predictions are obtained 1 hour into the future. When using the BMC model to correct the occupancy count of the raw sensor readings using the particle filter in Chapter 4, predictions were done 1 second into the future.

Before we discuss actuation, we must again discuss American Society of Heating, Refrigerating and Air-Conditioning (ASHRAE) standards for thermal comfort. ASHRAE Standard 55 [ASH07a] uses the predicted mean vote (PMV) metric to establish levels for occupant comfort. This metric incorporates multiple parameters such as humidity and airflow to estimate occupant comfort on a continuous scale from -3 to 3, where negative values indicate slightly cool to cold (-1 to -3) and slightly warm to hot (1 to 3). The optimal temperature corresponds to PMV = 0. ASHRAE states that acceptable room conditions are temperatures such that $-0.7 \leq PMV \leq 0.7$. For our deployment we choose more conservative

Algorithm 5 Actuation Algorithm

```
$CondTemp_{i,j} \leftarrow$ Room temp from time $i$ to $j$
$CurrHour \leftarrow$ Current hour
$T_{TG} \leftarrow$ Temperature such that $PMV = 0$
$T_{ASH} \leftarrow$ Temperature such that $-0.5 < PMV < 0.5$
$pThresh \leftarrow$ Probability threshold of occupancy

$BMC(CurrOcc, predLen)$
Returns probability vector when occupancy is likely

$ThermalDelay \leftarrow$ Time to reach $T_{TG}$ given $T_{ASH}$

- Program Start -
for Every $n$ minutes do
  $CurrOcc \leftarrow$ Current Particle Filter estimate of occupancy
  $occPred \leftarrow BMC(CurrOcc, predLen)$

  for Each room $r$ and point of time $t$ in $occPred$ do
    $occupied \leftarrow$ Periods $occPred_{t \rightarrow t+60} > pThresh$
    if $occupied > 5$ minutes of next 15 minutes then
      $CondTemp_{t-ThermalDelay,t+15} = T_{TG}$
    else if $5 \leq CurrHour \leq 24$ then
      $CondTemp_{i,i} = T_{ASH}$
    end if
  end for
end for
```

temperatures where $-0.5 \leq PMV \leq 0.5$.

Algorithm 5 shows how we combine the current occupancy determined by the particle filter, predicted occupancy determined by the transition model, and the estimated thermal ramping time (estimated from historical data) in order to schedule when to begin conditioning. There are two temperature set-points; T_{TG} and T_{ASHRAE}, which are the target and ASHRAE set-points respectively. The algorithm's purpose is to change these set-points optimally based on room usage. Every n minutes (15 minutes for our deployment) we check the current occupancy state of the rooms. We then make a prediction *predLen* (60 minutes) into the future using the transition model. We examine each the predicted occupancy probability for each room. If we find a window of time where the room will be occupied 5 minutes out of the next 15 minutes, we use the experimentally determined thermal delay (time required to reach the target temperature), and schedule the set-point to the target temperature. Otherwise, we condition to the ASHRAE set-point. This ensures the room will be at a reasonable temperature if a prediction is false.

7.3 POEM System Evaluation

There are two aspects of the system to be examined. The first is that rooms are conditioned appropriately given actual occupancy usage. The second is to determine how much energy could be saved when applying the strategy to the floor of a building.

We tested two versions of POEM in a live deployment. The first strategy uses only PIR for occupancy detection and the same actuation using binary occupancy data. PIR data does not have issues with cumulative error, hence a particle filter is not used. Since PIR cannot determine how many people are in an area, ventilation control is done assuming maximum occupancy; an occupied room is assumed to

		Ramping
	Area	Duration
Conference Room	40 m^2	45 min
Office Area 1-9	36 m^2	30 min
Computer Lab	111 m^2	60 min
Conference Room	40 m^2	30 min
Dean's Conference Room	52 m^2	30 min
Robotics Lab	18 m^2	30 min

Table 7.1: Zone information for deployment. The ramping duration is the time required to reach the target temperature from the setback temperature.

be fully occupied. This is equivalent to baseline ventilation strategy. This is common practice since HVAC designers and building managers tend to be over cautious with respect to ventilation. Indeed we found many areas that were over-ventilated with respect to even maximum occupancy. The second strategy uses OBNet (Camera and PIR data) along with the particle filter. Ventilation in this case is done according to ASHRAE 62.1 [ASH07b]. Each strategy was tested for two weeks for 16 zones (2 AHUs), accounting for 30% of the area that the AHUs condition. Approximately 52 people occupy these areas and occupants in the area were informed of the camera use with the experiments and the data collected during our experiments has been handled following best practices [OEC, eur].

Table 7.1 shows size of each zone and how long each zone takes to get to temperature from the setback temperature.

To evaluate the performance of the system over the course of a year, we developed a calibrated EnergyPlus simulation that closely matches the performance of the actual building. This was done using 6 weeks of historical data from the building from different seasons; 2 weeks Fall, 2 weeks Winter, 2 weeks Summer. Each day was compared with a similar day from the EnergyPlus simulation with respect to weather/temperature. On average, the EnergyPlus simulation had a normalized RMSE of 6.2%. With a calibrated model, we are able to accurately determine the performance of these strategies under exactly the same conditions; without a model, we would be required to run each system for potentially years for enough data for a valid comparison. Using the calibrated model, we test four different strategies. Two strategies tested are the same PIR and Camera/PIR strategies as the live deployment. In addition, we test a reactive strategy where rooms are conditioned immediately once occupied without predictive temperature ramp up. Similar to the predictive PIR approach, we assume maximum occupancy when the room is occupied for ventilation purposes. The last strategy is an "Oracle" strategy, where we condition the space assuming perfect prediction and perfect measurement of occupancy. The oracle strategy gives us an upper bound of the energy savings that is possible if the Camera/PIR system were to run with perfect information.

7.3.1 Energy Savings

Our building is conditioned from a centralized chilled water tower and boiler systems. The chilled water tower stores water within a tank large enough for stratification to occur; warmer water naturally rises to the top and cold water falls the bottom of the tank. Cold water is supplied by pumping water at the bottom of the tank. Additional cooling is supplied to the tank as necessary. The hot water is supplied by several hot water boilers. The chilled water tower boilers supply the hot/cold water to air handler units (AHU) and variable air volume units (VAV) for heating and cooling. The AHU is used for air cooling the air. The VAV units (Figure 7.3) are used for heating and regulating cool air to individual zones. This is done by using a damper to regulate the air volume that passes over heating/cooling coils. For our deployment, we are interested in how much energy the VAV units and AHUs are consuming.

The amount of energy lost/gained by the coil is equal to the amount of energy

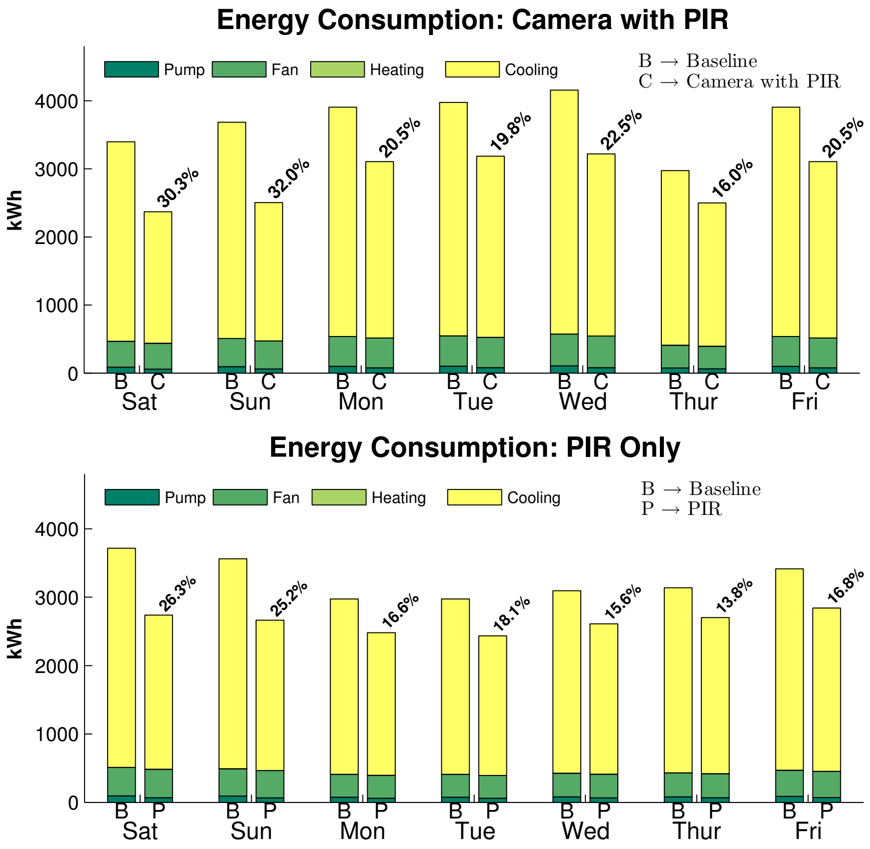


Figure 7.2: Energy consumption breakdown for 7 representative days. PIR and camera/PIR based POEM saves on average 21.1% to 26.0% respectively.

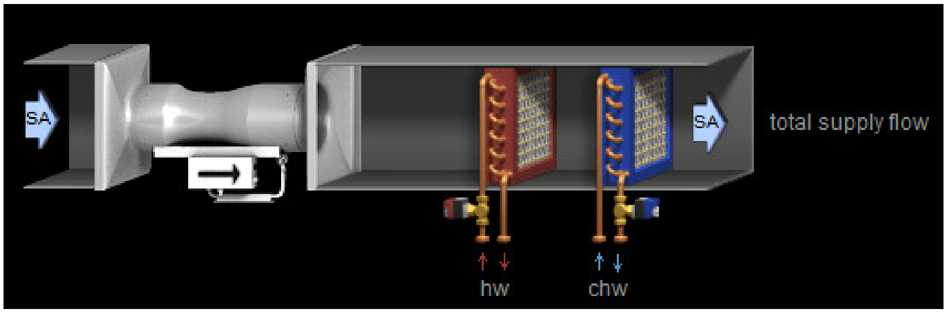


Figure 7.3: Supply air (SA) flows through coils, heating or cooling the air.

lost/gained by the air. We can estimate the energy consumed by a VAV or AHU using the heat balance equation,

$$Q = mC(T_{sa} - T_{da}) \tag{7.1}$$

where Q is energy transferred to the air, m is the total mass of the airflow over an interval of time, C is the heat capacity of air, T_{sa} is supply air temperature, T_{da} is the discharge temperature of the air after passing through the coil. By measuring the airflow and the supply and discharge temperatures, we can calculate the energy consumed.

7.3.2 Live Deployment Results

The baseline strategy of the building is a static schedule where the room HVAC system (temperature/ventilation) is turned on at 6 am and then shut off at 1 am the next day. From 1 am to 6 am, room temperatures "float". In order to measure the amount of energy consumed by the baseline strategy, we measure the energy consumption of the system without POEM running. Days with similar weather and temperature profile are used for the baseline comparison.

Figure 7.2 shows energy savings and breakdown for one particular week for both strategies. The left and right bar for each day shows the baseline and strategy breakdown respectively. Above each strategy, we show the percentage savings over the adjacent baseline bar. Since the experiments were conducted during the summer, no zone required heating. In general, the energy from the supply and return fans along with the hot/cold water pump remained fairly constant. Overall PIR and Camera/PIR saved on average 21.1% and 26.0% respectively. For both strategies, the most savings occurred on the weekend where many zones were unoccupied. For weekends, the PIR only strategy saved 25.1%-27.5% and the Camera/PIR strategy saved 30.2%-32.0%. On the weekdays, PIR saved 13.8%-18.0% and the Camera/PIR strategy saved 16.0%-22.5%. For areas that are consistently

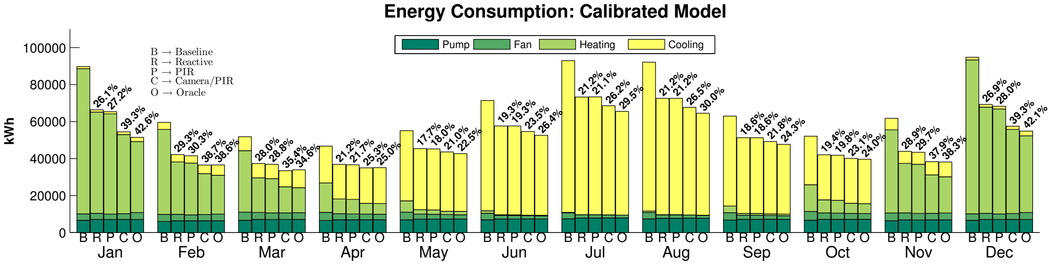


Figure 7.4: Energy consumption based on calibrated energy simulation. On average PIR and Camera based strategies save 24.5% and 31.0% annually respectively.

occupied with regular schedules, the savings were achieved by turning off the system earlier than the 1 am shutoff time, or delaying the ramp-up in the morning. The difference between these strategies is due to ventilation rate. Since the Camera/PIR strategy is able to reduce ventilation rates, this strategy is able to save more energy than predictive PIR strategy. For other areas not consistently occupied, such as the conference room, energy savings were possible in the morning where the temperature ramp-up could be delayed and in the evening when the room was unoccupied.

The Camera/PIR strategy saves more energy since ventilation is based on the number of people whereas the PIR only strategy ventilates based on the maximum occupancy. Ventilation has a significant impact on energy consumption since increasing ventilation increases the amount of outside air to be conditioned. This is similar to trying to cool a house down on a hot summer day with the front door open.

7.3.3 Calibrated Model Results

As previously mentioned, we also test four different strategies using the calibrated model. Figure 7.4 show the energy savings and breakdown for each strategy. When we compare the real world results to the calibrated model results, we see similar energy savings for the periods of similar weather. For the predictive conditioning, we saw 21.1% average savings for live deployment and see 19.3% for the calibrated model. We also see a similar difference between the PIR and Camera/PIR strategies for the calibrated model as compared with the live deployment. For the live deployment we saw difference of 4.9% (Camera/PIR 26.0% vs PIR 21.1%); the calibrated model showed a difference of 4.2%. These results supports our initial results (normalized RMSE of 6.2% between model and historical results) and that our calibrated model is able to accurately measure energy differences between strategies.

When comparing the predictive PIR and reactive PIR strategies, we see similar results; we only see a maximum difference of 1.1% during the coldest months (Dec, Jan). A priori one would expect the predictive strategy to use slightly more energy since it conditions spaces over longer periods of time due to the fact that is trying to anticipate expected occupancy (i.e. load). Interestingly, the predictive PIR has slightly higher energy savings. The predictive PIR indeed conditions a larger total amount of time than the reactive PIR strategy; on average the predictive PIR conditions on average 1.2 hours more per day per room compared with reactive PIR. However, the temperature differential from the initial to the final conditions for any period of occupancy is on average smaller for predictive than the reactive strategy. On average the temperature difference between the target and actual room temperature when occupants initially occupy a room is approximately 0.08 C^o (0.15 F^o) for predictive PIR and 1.7 C^o (3.0 F^o) for reactive PIR. The reactive strategy has to work harder to ramp up the room temperatures between periods of occupancy; in some cases, it is more energy efficient to maintain temperature than to ramp up temperatures from a very low level. As we'll see in the following section, this also dramatically impacts temperature effectiveness as perceived by the building's occupants.

The predictive PIR and the Camera/PIR strategies show 24.5% and 31.0% annually respectively, with the largest savings occurring during the coldest months.

The main cause of this difference is the ventilation. We also see how ventilation plays a major role in the energy savings when comparing the Camera/PIR strategy to the Oracle strategy. In general the predicted schedules from the Camera/PIR strategies closely match the Oracle strategies (3% difference). One major difference, which is discussed in more detail in section 8.4, is that the Camera/PIR has additional ventilation added in order to account for possible errors in the occupancy count. When comparing Oracle and Camera/PIR results, we see that the Oracle saves 33.0% whereas Camera/PIR saves 31.0% overall. In some instances, the Camera/PIR saves slightly more than the Oracle. This occurs when the Camera/PIR method predicts rooms to be empty space when in reality the space is occupied. Further analysis of the energy traces shows that this difference is indeed due to over-ventilation by the later due to the small uncertainty in occupancy (i.e. a safe guard band added it by design to cope for potential under-counting). This can be seen in Figure 7.7 where we see that the PIR/Camera strategy is above the required ventilation rate for the majority of the time.

7.3.4 Conditioning Effectiveness

In this section, we examine the conditioning performance of POEM. We are interested how close operational temperatures were to the target temperature during times of occupancy and how effectively we could reduce ventilation rates using occupancy.

To evaluate the thermal effectiveness of POEM, we calculate the root mean squared error (RMSE) between the target temperature and the observed room temperature during the periods of when the room is occupied. For this analysis, we will only examine the Camera/PIR strategy as we only gathered ground truth data for the days of the Camera/PIR deployment. It is difficult and time consuming to process ground truth and was too labor intensive to process additional days. In order to have a basis of comparison, we also consider the ability of the

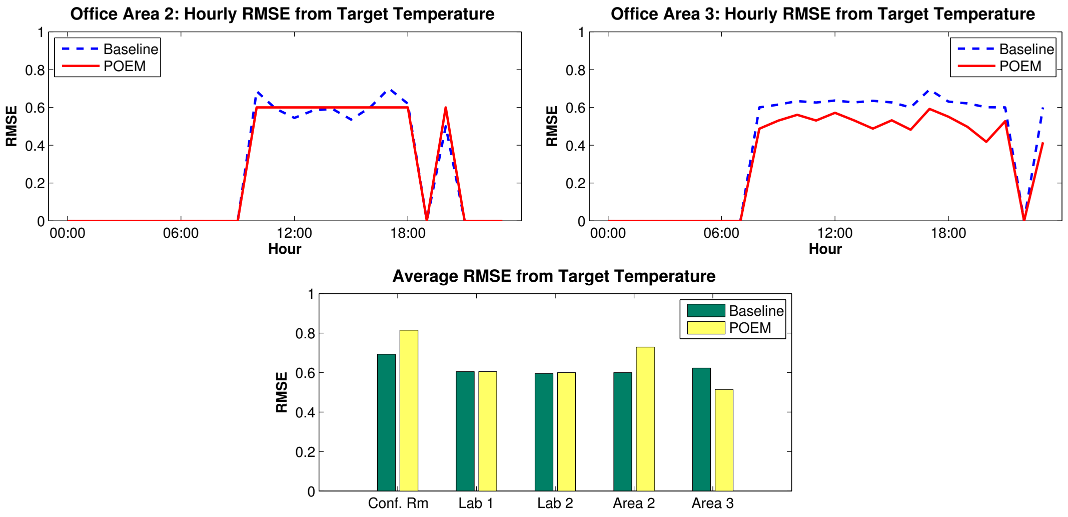


Figure 7.5: RMSE (F°) between the target and observed room temperatures

baseline strategy to meet target temperatures. In most cases, the baseline temperature deviates approximately 0.6 C° (1 F°) from the target temperature. This is expected as the proportional-integral-derivative (PID) controller of the VAV will cause the temperature to oscillate with a small amplitude from the target temperature. Figure 7.5 shows the overall RMSE for each room. Of particular interest is the temperature of the conference room as this room does not have a consistent schedule and most likely to deviate from target temperatures. In this case, the RMSE of POEM is on average 0.25 C° (0.45 F°) whereas the average RMSE of the baseline strategy is 0.21 C° (0.38 F°). As this is only a difference of about 0.07 C° (0.12 F°), we can see that the conditioning effectiveness of POEM is close to baseline. This room tended to be farther from the target temperature for both baseline and POEM strategies since the conference room has two exterior walls and receives solar gain. For areas with regular schedules, we found POEM's ability to condition comparable to the baseline strategy; in most cases, the difference between POEM and baseline RMSE is less than 0.06 C° (0.1 F°).

In addition to the actual results we also examined the thermal effectiveness of the reactive and Camera/PIR strategies from the calibrated model. Figure 7.6

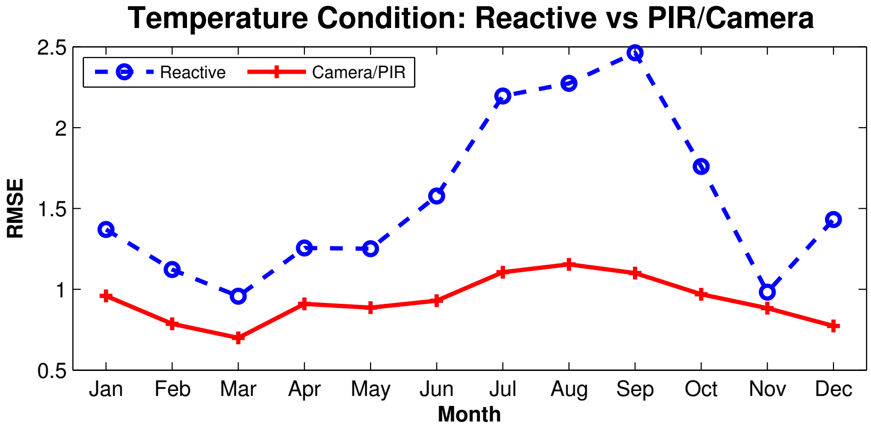


Figure 7.6: RMSE (F°) between the target and observed room temperatures from the calibrated model.

shows the RMSE for each strategy for the conference room. The Camera/PIR has substantially better performance than the reactive approach; the Camera/PIR approach has an average RMSE of 0.52 C° (0.93 F°) whereas the reactive approach has an average RMSE of 0.83 C° (1.5 F°). This is expected since time is required for rooms to ramp to the target temperature. This is especially true during the warmer months where solar gain can greatly affect the ability of the HVAC system to meet load. During these months (Jun-Sept), the RMSE of the reactive system is 0.83-1.33 C° (1.5-2.4 F°), which can easily be perceived by occupants. Thus, the Camera/PIR approach is better able to meet the target temperatures.

ASHRAE standard 62.1 [ASH07b] requires the outdoor ventilation to be $V_{bz} = R_pP_z + R_aA_z$ where z denotes the zone, V_{bz} is the ventilation rate, R_p is the minimum CFM (1 CFM = 1 ft^3/min) per person, P_z is the number of people, R_a is the minimum CFM/ft^2, and A_z is the floor area. R_a and R_p changes depending on the target use of the room as specified by the ASHRAE 62.1 standard. This standard is for areas without a demand ventilation system. For our rooms, the required CFM per person is 5. Examining Figure 7.7, we can see that a typical baseline ventilation strategy is far more inefficient than POEM. Office

1 under-ventilated 1%. During these periods of under-ventilation, the average under-ventilation was 2.5 CFM. As 5 CFM is required per person, the magnitude of the under-ventilation is not large. The lab is theoretically under-ventilated an average of 2.98 CFM 4.8% of the time. When examining the actual CO_2 levels, we found the areas were properly ventilated. In order to evaluate proper ventilation, we examine the CO_2 levels with respect to the ASHRAE 62.1 demand response ventilation standards. This standard states 925 ppm is considered acceptable for office spaces. Figure 7.7 shows the CO_2 levels for the most densely occupied area in the deployment, which is the conference room. For the day shown, several groups meet back-to-back. From the figure we can see that the levels never exceed 925 ppm showing that adequate ventilation is being provided. We can also see that the Camera/PIR strategy has slightly higher levels since the strategy provides less ventilation, but is well under the threshold (600 ppm vs 925 ppm). This suggests the ventilation provided by the previous equation is above what is actually required.

7.4 Return on Investment Analysis

In this section, we use our results in order to estimate whether deploying POEM is economically viable. We extrapolate our results to the entire building, which contains a base area of 6,689 m^2 and is primarily used for research labs and offices. The building has an annual energy consumption for conditioning of 5,275,992 kWh electrical power and 246,000 therms of gas. The cost for electrical energy depends on the time of a day and season, it is usually between $0.13 and $0.18 per kWh, the price for gas is constant over the year at $0.7 per therm of gas. All these prices are special rates negotiated with a utility company. For reference, the same company charges $0.34 and $0.22 per kWh for residential and commercial customers. For an average price of $0.15 per kWh electrical power and $0.7 per therm gas the

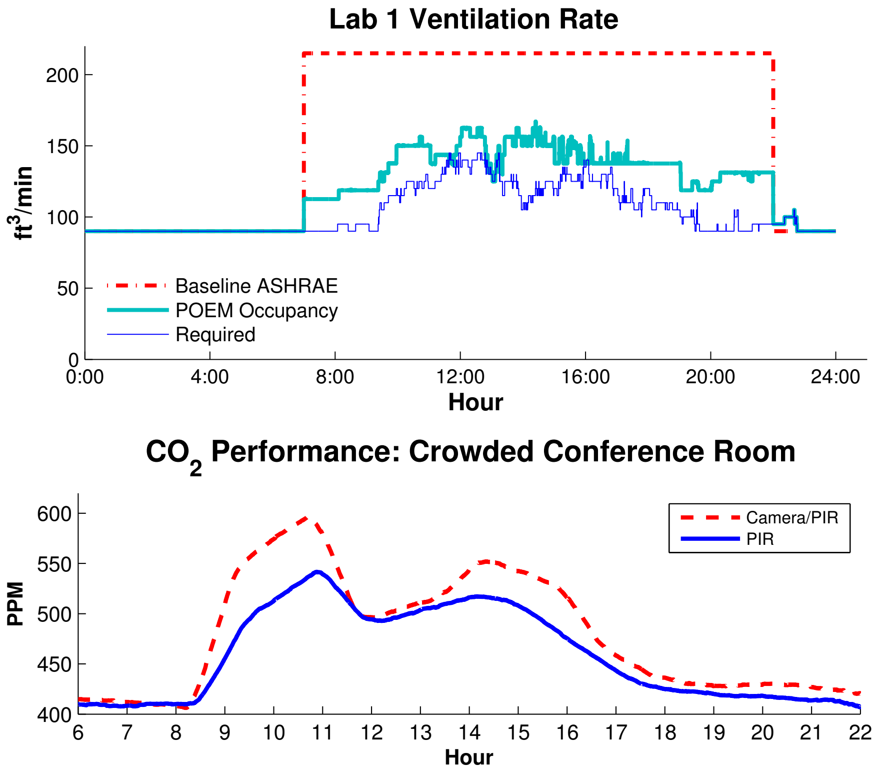


Figure 7.7: Ventilation and CO$_2$ levels

annual conditioning costs of the mentioned building are $963,598.80. One node of our system consists of the components listed in Table 1, which sums up to a price of $710 for camera and $215 for PIR nodes. This cost can be reduced through commercial production.

We estimate that each node costs $35 to deploy, and $8k is required for testing the system. For a 3 story 6,689 m^2 building, about 65 camera nodes and 360 PIR nodes are required to provide complete coverage. Therefore manufacturing and deploying the system would cost approximately $147k. Savings of 26% of $963k equals to $250k per year or $20.8k per month. The largest ongoing cost for our system is battery maintenance. A conservative estimation of the energy consumption of a camera node is about 80mW; with three battery packs with a capacity of 11 Ah each, the battery packs have to be exchanged and recharged every 60 days. Since the energy consumption of a PIR node is low compared to a camera node, a PIR node is able to run 175 days on one battery pack rather than 60 days. This results in a maintenance cost of about $30 per camera node per year and about $10 per PIR node per year. For the whole 65 camera and 360 PIR node system, this would cost $5,500 for maintenance each year. Depending on the market price for energy, the manufacturing and deployment costs of our system would be amortized between 6.1 and 8 months. The energy savings of a system with PIR sensors only are about 21.1%. This would result in an average amount of $203k for energy savings and an amortization time between 4.8 and 6.4 months due to the reduced system costs of $94k and maintenance costs and $3,800 per year. As shown in Figure 7.8, a system consisting of only PIR sensors has a shorter amortization time compared to a system with camera and PIR sensors. Since buildings are usually operated for many years (70-75 years [bed11]), in the long term, a combination of camera and PIR sensors always achieves higher savings.

These results also show that despite the extra work of adding cameras, the savings are significant to justify the additional effort. This is due to the magnitude

Table 7.2: Prices of camera and PIR nodes

Component	Price	Component	Price
Imote2	\$350	Tmote	\$120
IMB 400	\$250	PIR sensor	\$30
Batteries	\$100	Batteries	\$60
Fisheye lens	\$5	Assemble (10 min)	\$5
Assemble (10 min)	\$5		
	\$710		\$215

of the money currently spent on energy. Though the difference of the energy savings between PIR and Camera/PIR is only 5%, this still constitutes a non-trivial amount of money. For perspective, over a 5 year period, the Camera/PIR system saves approximately \$140k-\$200k more than PIR alone (see Figure 7.8). Over the average lifetime of the building, which is approximately 70 years, the system can potentially save \$15.0M to \$20.0M.

7.5 Lessons Learned

We found that deploying, testing, verifying the performance of the system to be extremely difficult. For the camera system testbed, a customized Imote2 debug board was designed and manufactured for each of the 22 cameras. For the PIR sensors, 40 nodes had to be hand "manufactured" and deployed after negotiating with facilities, staff and students. In particular, facilities required us to have CO_2 monitoring before we could proceed. In order to achieve this, we developed a wireless network of 10 CO_2 nodes using Tmotes and CO_2 sensor boards.

One of the most time consuming part was the processing of ground truth data. We tried multiple techniques to speed up the processing of the image data. Initially, we tried using state-of-the art techniques that counted torsos

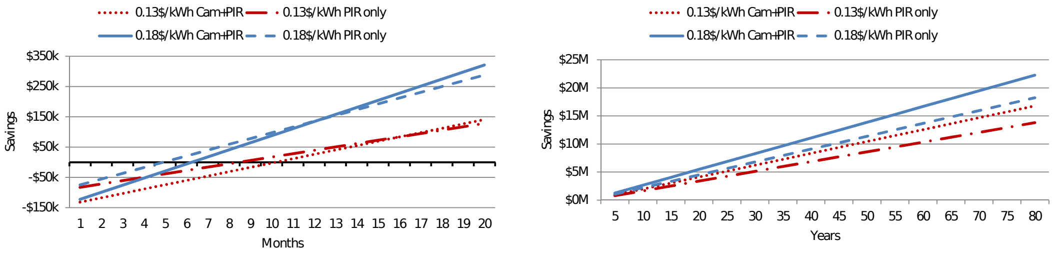


Figure 7.8: The left compares savings for different energy prices. The right shows savings over 80 year period.

and legs [Ope, VJ01, LM02] in order to detect the presence of humans. We found even the best techniques were only about 80% accurate and in the end found background subtraction a much more reliable method of identifying images with occupants. We also tried various types static background techniques but found that when people loitered of left static objects such as chair and boxes, this would dramatically increase the number of images needed to be manually examined. The most effective method we found for detecting transitions is similar to the one presented previously. We use previous frame background subtraction in order to identify images where objects are moving and then manually process this data. After applying this technique to the images, each camera produces 1000-3000 images per day that potentially contains transitions. Processing the ground truth required examining approximately 150,000 images.

Despite having deployed in a new building with a BMS, we experience multiple difficulties with regards to instrumentation. For several zones, we found that during construction, temperature sensors were left out of VAV boxes. This required us to deploy commercial wireless temperature sensors so we could measure the exhaust air temperature; this was accomplished by examining the plans, locating the VAV boxes, obtaining permission from facilities to install sensors, and physically drilling duct holes and mounting the sensors. Calibration of the duct flow sensors was also an issue. While testing the system, we found the AHU flow measurements to be faulty with respect to other air flows measurements. For several months, multiple calibrations were performed and some flow sensors had to be replaced by facilities. Flow sensor measurements need to be verified prior to deployments.

7.6 Summary

In this chapter, we designed and implemented POEM, a close loop system that conditions rooms based on occupancy on a real production building. POEM uses two wireless sensor networks developed in Chapter 4; one network of cameras called OPTNet, and another network of PIR sensors called BONet. By opportunistically controlling the HVAC system based on occupancy, we showed savings of 26.0% are possible while maintaining conditioning effectiveness. Using a calibrated model, we estimate that $\approx 30\%$ savings are possible annually. Given the cost of heating and cooling, we show that these savings would amortize the POEM system within about 6 to 10 months. By providing greater value from the same physical plant, our approach can move beyond cost-to-build and cost-to-operate metrics to broader return-on-investment for new technologies. In the next chapter, we test our control strategy in a live deployment except using the thermal based occupancy sensing system from Chapter 5.

CHAPTER 8

OBSERVE: ThermoSense

In the previous chapter, we tested a system utilizing the control strategy from Chapter 3 developed around the camera based occupancy system from Chapter 4. In this chapter, we test the thermal based occupancy system from Chapter 5 using the OBSERVE strategy from Chapter 6 in a building energy simulation. Live actuation of the building using ThermoSense was not possible at the time and is left as future work. We use the ThermoSense system to collect 3 weeks of occupancy data from the target building and used the actual building energy to calibrate the energy simulation to match the target building.

In this chapter, we contribute the following:

- The design, implementation, deployment and evaluation of a full close loop system that conditions rooms based on occupancy measured using the Thermosense system.
- We tested four different strategies for HVAC conditioning and show how different sensing and actuation strategies affect energy consumption and occupant comfort. We showed that by using this system for conditioning usage based control of temperature and ventilation we can save 25% energy annually while maintaining occupant comfort.

8.1 Energy Analysis

With the occupancy estimation system in place, we next would like to use data collected from the ThermoSense system to estimate potential energy saves using different strategies. Since KNN had the best performance in terms of RMSE and NRMSE, as seen in Chapter 5, we use this model for our energy analysis. In this section, we evaluate four strategies to reduce energy usage and compare them to a baseline strategy that operates on a static schedule. We use the OBSERVE strategy described in Chapter 6. The OBSERVE method utilizes a Blended Markov Chain (BMC) that continuously updates its prediction based on the current occupancy estimate. If trained with binary data, this model can give binary predictions of occupancy. With discrete training data, the model is able to predict discrete levels of occupancy. Based on a probability threshold, the room is conditioned beforehand if the room is likely to be occupied in the coming hour. Ventilation is controlled according to ASHRAE 62.1 [ASH07b]. For our analysis we use predictions discrete occupancy, binary predictions, and a purely reactive strategy using no prediction.

We trained the BMC's using 3 weeks of data collected from the ThermoSense system deployed in 8 offices, 1 lab and 1 conference room in an academic commercial building. Using the BMC, we generated 7 days of simulated occupancy. To simulate system error, we introduced error based on a normal error distribution. This distribution was determined by examining the distribution of the residuals from our RMSE analysis and performing a normal fit. The frequency of these perturbations was added using an exponential distribution. We found that the duration between the errors were exponentially distributed.

To test these strategies, we utilize EnergyPlus [Epl], which is a state-or-the-art industry standard tool for simulating buildings. With this simulator, we are able to change occupancy over time to determine how these strategies affect efficiency.

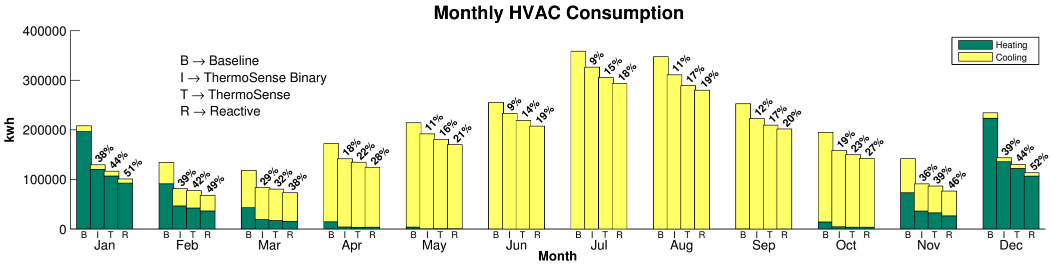


Figure 8.1: Monthly heating and cooling consumption for the different strategies.

By using a simulator, we are also able to test the strategies under identical weather conditions. In our simulation, we used materials similar to the actual building and sized rooms to match the deployment. Target set-points of 24C° (75.2F°) and 20C° (68F°) was used for cooling and a cooling and heating set-points respectively. Since our deployment only covered part of the entire building, we assumed the same schedules for the rest of the building in the simulation. The simulated commercial building contains offices, labs, and meeting rooms. The location of the simulation was in the central valley of California. Other locations were not evaluated in the interest of space.

Figure 8.1 shows the monthly breakdown of the heating and cooling. All three strategies had significant energy savings over the baseline strategy. Overall, the reactive strategy had the lowest energy consumption. This strategy consumed 29.6% less than the baseline strategy annually. It saves additional energy since it does not pre-condition rooms ahead of time. However, we will see that this savings in energy comes at the cost of comfort (Section 8.2). In general, all the strategies had the largest percentage of savings during winter months (Nov - Feb) and the lowest percentage of savings during the warmer months (May - Sep). ThermoSense and ThermoSense Binary had 24.8% and 19.7% savings respectively. The binary approach consumed more energy since this strategy has a tendency to over-ventilate. Over-ventilation reduces efficiency since it increases the amount of outside air that needs to be conditioned. The greatest differences between the

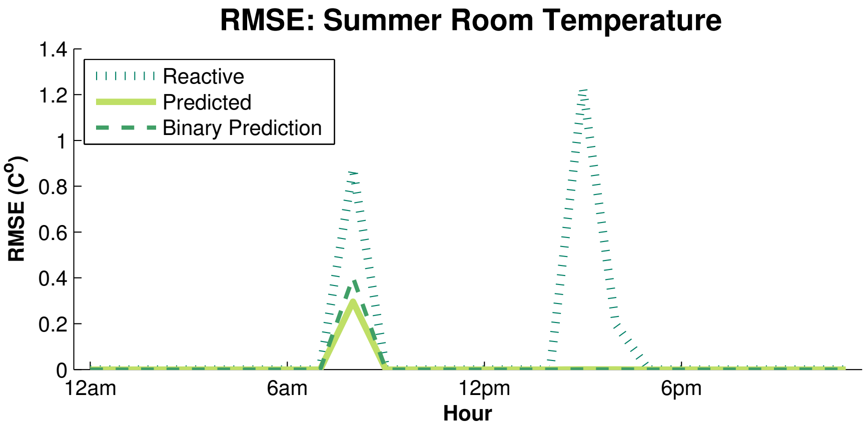


Figure 8.2: The temperature RMSE for the periods when the room was occupied during the summer.

ThermoSense and ThermoSense Binary approaches occur during the warmest and coldest months (Jan, Dec, Jul, Aug) where ThermoSense consumed 5-6% less than ThermoSense Binary. Increased ventilation of ThermoSense binary during these months has a greater impact on efficiency than during milder shoulder seasons.

8.2 Conditioning Effectiveness

In the previous section, we show that occupant based conditioning saves a significant amount of energy. However, conditioning effectiveness also needs to be considered.

For the temperature effectiveness, we examine a room that is only occupied at 8am, 3pm, and 4pm on Mondays and focus our analysis to the warm months (May - Sep). For our analysis, we examine the RMSE of the room temperature from the target temperature for each hour. Figure 8.2 shows the RMSE for each hour of the day. For un-occupied periods, we consider the RMSE to be 0. The reactive strategy had the worst overall performance. At 8am, the RMSE was 0.877C^o during the hour that the room was occupied. At 3pm the room had

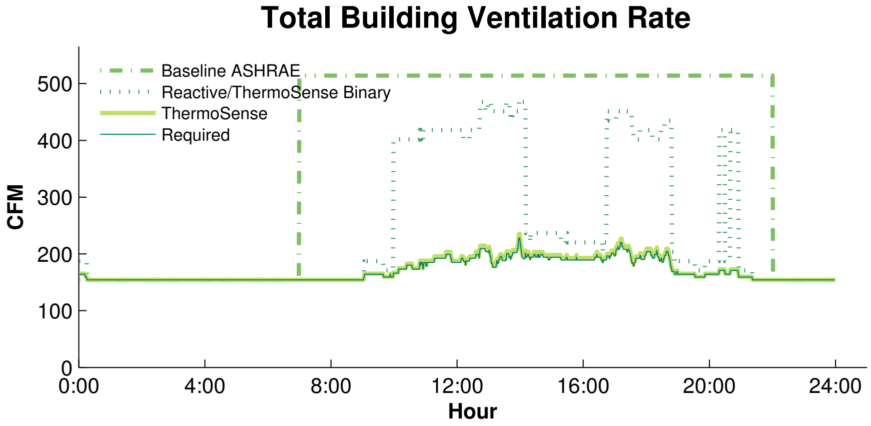


Figure 8.3: Ventilation effectiveness of the strategies.

a RMSE of 1.23C^o. This higher RMSE is due to the solar gain increasing the temperature of the room. However, at 4pm the RMSE drops to 0.197C^o. In this case, the occupied state of the room at 3pm carried over to the 4pm period. Thus, while some energy was saved by not preconditioning, we can see that this was at the cost of thermal conditioning. Both ThermoSense prediction strategies perform substantially better than the reactive strategy. At 8am the ThermoSense and ThermoSense Binary prediction methods had RMSE values of 0.309C^o and 0.415C^o respectively. At 3pm-4pm, both prediction strategies had nearly identical RMSE values; both were close to 0.033C^o. Again, while the ThermoSense prediction strategies consumed slightly more energy, they did not compromise thermal conditioning.

Next we examine the ventilation effectiveness of the system. For our analysis, we evaluate the ventilation of the spaces for one particular day. As previously mentioned, we use the ASHRAE 62.1 standard to determine the ventilation according to occupancy. As ventilation is regulated in real-time, the binary based strategies (reactive and ThermoSense binary prediction) have the same ventilation rates. As we cannot determine the precise occupancy with a binary occupancy estimate, we assume maximum occupancy for rooms that are occupied for ventilation. The baseline strategy assumes maximum occupancy from 7am to 10pm. Figure 8.3 shows the baseline, ThermoSense binary, and ThermoSense based strategies. We can see that the baseline strategy greatly over-ventilates. However, there is also a short period at the beginning where the baseline strategy under-ventilates; this illustrates the shortcomings of a static schedule. Both the ThermoSense binary and ThermoSense strategies were able to meet the required ventilation level. However, we can see that the ThermoSense binary strategy still over-ventilates the area a great deal; overall, this strategy over-ventilated the area by 170%. The ThermoSense binary strategy performed best when multiple rooms were empty, which can be seen during the period between 2pm and 5pm. Since only a few rooms were occupied, only a few rooms were over-ventilated. The ThermoSense based ventilation performed the best. We can see that the ventilation rate is only 3.25% more than the required rate. This is due mainly to a 10% occupancy increase added by design in order to protect against under-ventilation that could be caused by sensor errors, as shown in Chapter 6.

8.3 Summary

In this chapter, we tested a occupancy based control strategy utilizing the Thermosense system we developed in Chapter 5 and the control strategy from Chapter 6. We tested the control strategy in with 17-node deployment covering 10 building conditioning areas totaling 2,100 sq. ft. for a period of three weeks. We tested four different usage based conditioning strategies and analyzed the energy usage; we showed that 25% annual energy savings are possible with occupant based conditioning strategies. These results are consistent with the findings from our live camera based system of the previous Chapter. Thus, we are confident that the ThermoSense system is at least on par with the system described in the previous chapter and with the added benefit of avoiding issues related to the cumulative error problems of a turnstile based system (Chapter 4). In the next chapter, we next explore occupant driven strategies to increase thermal comfort.

CHAPTER 9

Thermovote

Thus far, we have developed strategies that save energy while minimizing the discomfort of occupants. In this chapter we now instead focus on increasing the occupant thermal comfort level. In recent years, multiple works have been published showing how wireless sensor networks can be utilized to reduce the energy consumed by buildings[ECC11, ABD11, LSS10]. However, within the WSN community, there has been little research attempted to improve the quality of service for users. Instead, the aim has been simply to maintain, or in some cases potentially decrease quality of service in order to achieve greater efficiency. While increasing efficiency is an important goal, the service that HVAC systems provide is arguably more important than reducing energy. Before we attempt to reduce energy for HVAC systems, we must first ensure that the system is providing adequate service.

While single occupant offices are simple to achieve thermal comfort, the task becomes more difficult when multiple occupants share the same temperature set-point. Building managers typically rely on the sensor networks of building management systems (BMS) to maintain user comfort. Managers choose a single temperature set-point for occupied periods. This set-point is typically chosen based on the criteria set by American Society of Heating, Refrigeration, Air-Conditioning Engineering (ASHRAE) Standard 55 [ASH07a]. This standard uses Predicted Mean Vote (PMV), to estimate the most comfortable temperature. PMV uses multiple parameters such as humidity, temperature, and air flow to estimate how warm or cold occupants feel on a discrete scale from -3 to 3. Positive values indicates occupants are warm and negative values indicates occupants are cold. A PMV of 0 indicates occupants are comfortable. Temperature set-points are chosen by assuming fixed values of most parameters and then solving for a temperature will give a PMV value of 0.

Estimating PMV is inherently prone to error. Often, many of the values for parameters such as clothing coefficients or activity levels are given fixed values based on tables supplied by bodies such as ASHRAE. The clothing coefficient has been shown to differ by up to 20% depending on the table and method [BFB93] [ASH07a] [ISO06]. Other parameters such as mean radiant temperature is currently not measured by most BMS systems and is again often estimated [RA09]. While it is possible that PMV estimates could be improved by attempting to measure the parameters such as occupant activity, perceived airflow, and radiant heat for each space, this would require additional cost in terms of installation and development of specialized sensors. Even if perfect measurement is possible, the differences among individual preferences make error in the PMV estimate inevitable.

Rather than develop and deploy an entire testbed dedicated to sensing parameters correlated to thermal comfort, we propose a more direct means of measuring occupant thermal comfort; ask the occupants. For this application, humans are the best sensors available. As data muling is to using humans for transport of data, participatory sensing is to using humans to sense data. We propose leveraging existing wireless infrastructure such as cell phone and use humans as sensors.

By controlling room temperatures directly using occupant feedback, we are more likely to increase the overall thermal comfort. This is already done in private offices where individual thermostats are available. However, for rooms with multiple occupants, it is not straight-forward to regulate temperatures from a single thermostat control such that the comfort level of the majority is maximized. Indeed, many spaces with multiple occupants have a disabled thermostat, a physically locked thermostat, or no thermostat control at all. In our building, which was constructed recently, only 6% of occupants had a "somewhat effective" thermostat. Occupants who do have a functional thermostat available must compete with each other or devise an equitable method of choosing an appropriate temperature.

This chapter contributes the following:

- We develop a crowd sourcing system that is able to gather thermal comfort data using a variety of platforms; iOS, Android, and webpage. Utilizing the system, we demonstrate two significant applications that leverage the collected system data.

- We develop an application around Thermovote for real-time building HVAC actuation based on user feedback. We also develop a method for removing error when calculating PMV by using occupant feedback that is more accurate than state-of-the-art methods of PMV parameter estimation. While other works simply gather data, our thermal control strategies (learned and real-time) are tested in real deployments for a significant period of time (15 and 25 months) with a large 135 subject population in two different buildings. Both user driven strategies performed significantly better than baseline and we show 67% and 80% satisfaction for the learned and real-time strategies respectively.

- We examine a second application that explores how the system can be used for the development of models of thermal comfort. We then demonstrate how these models can be integrated with building energy simulators in order to examine the effect that thermal preference adjustment has on energy consumption.

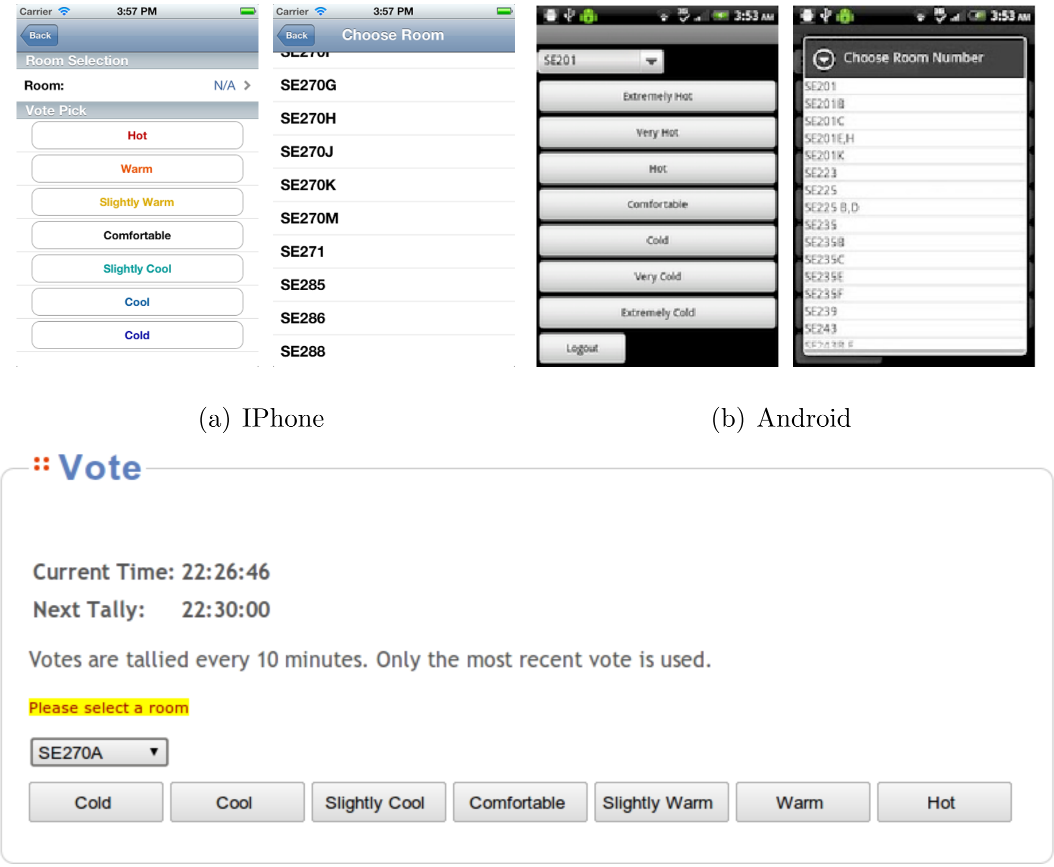
- We examine the relationship between occupant comfort and energy consumption by developing a thermal comfort based control strategy that regulates temperature based on a comfort range rather than a temperature range. We show that 13% can be saved annually by relaxing the acceptable comfort range to the limit of the standard. In addition, we explore different voting scales and show that a 7 point scale works best in general.

9.1 Thermovote Overview

Themovote is a system that occupants can utilize to report their current thermal comfort level. Users have the option of inputing their preference via cellphone, tablet, or website. Figure 9.1 shows the iOS, Android and web interfaces. Data from this system is feed into a database, which can then be used for a variety of applications.

9.1.1 Design Considerations

For our system, we gather thermal comfort data using the same 7 point scale specified by ASHRAE standard 55 [ASH07a]; hot (+3), warm (+2), slightly warm (+1), neutral (0), slightly cool (−1), cool (−2), and cold (−3). We use this scale rather than use temperature directly; temperature alone does not determine thermal comfort. Deciding what is a comfortable temperature is difficult. Occupants often have conflicting perceptions of what constitutes a comfortable temperature. As part of our surveys, we asked users what temperature they considered comfortable. We then compared this temperature to the actual comfortable temperature they were experiencing, we found a root mean squared error of 3.8 F^o; letting users choose a temperature will likely lead to discomfort. Figure 9.2 shows the variation in user opinion. Rather than using temperature, we decided to use the same 7 point scale as specified in ASHRAE standard 55 that ranges from -3 to indicate cold to +3 to indicate hot. While users may not agree on a numerical



Current Time: 22:26:46

Next Tally: 22:30:00

Votes are tallied every 10 minutes. Only the most recent vote is used.

Please select a room

(a) IPhone

(b) Android

(c) Website

Figure 9.1: The different methods of input provided.

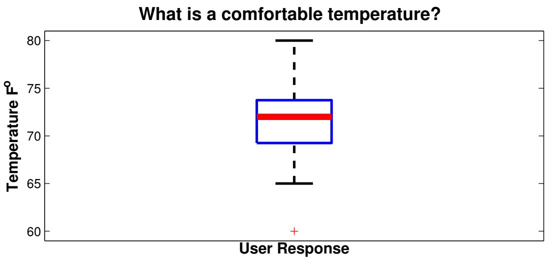


Figure 9.2: Variance of user opinion of what constitutes a comfortable temperature.

temperature, they are more likely to agree on whether they feel warm or cold.

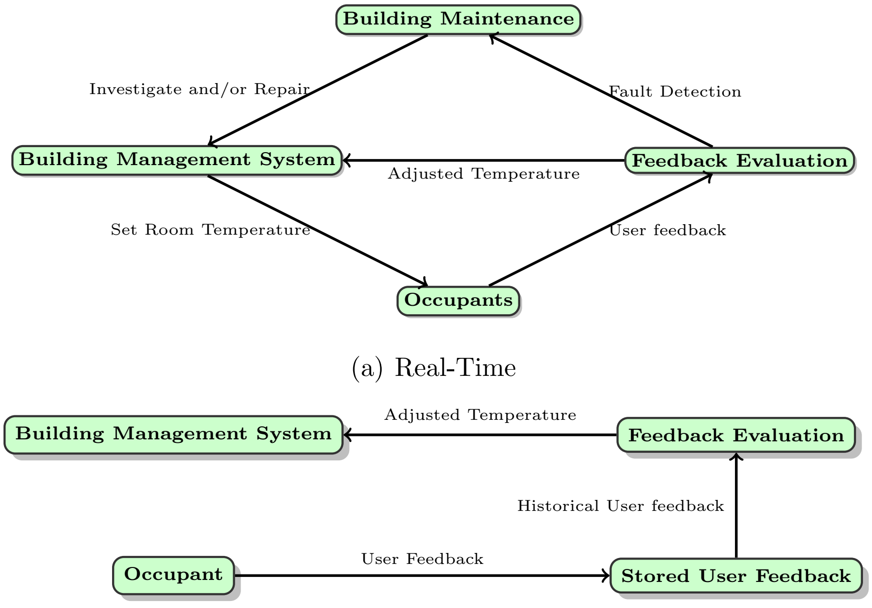
The next consideration is how to use this information. If a person states he/she is warm, then how much should the temperature be changed? Since there are many models of comfort developed, we decided to leverage off of the Fanger's PMV model (ASHRAE 55), which is one of the most widely used and studied models for thermal comfort. Typically these models are used to predict user comfort level. In our case, we know the actual comfort level. We can thus work backwards to solve for the parameters that will give a more comfortable temperature given user feedback. This is discussed in more detail in Section 9.2.

Lastly, we need to address how to handle diverging opinions. What happens if one person feels cold while another feels hot? To handle this issue, we use a voting based scheme. Users are allowed one vote valued from -3 (cold) to 3 (warm)for each voting period. After each voting period, votes can be then tallied and aggregated.

9.1.2 Systems

Figure 9.3 shows the feedback loop of our proposed system that utilizes a fully featured BMS such as ALC or Phoenix. Room occupants provide feedback regarding their thermal comfort level. This feedback is then evaluated. Depending on the feedback, the information is then used to adjust the room temperature, which is sent to the BMS and adjusts the room temperatures accordingly. In the case a fault is detected, building maintenance is notified to investigate the problem.

Since older buildings may not have a BMS capable of real-time changes, we also examine a strategy where we are only able to set a daily temperature schedule. User feedback is collected over a short period of time and data is stored in a database. This measured user comfort level data can be used to correct the error of the estimated PMV and determine an improved temperature set-point. This



(a) Real-Time

(b) Learned Control

Figure 9.3: Architecture of the Thermovote system. Here we show both real-time and learned versions of the system.

also allows managers to create different set-points throughout the day rather than have a single static set-point.

For our application, users provide one of the following “readings”; hot (+3), warm (+2), slightly warm (+1), neutral (0), slightly cool (−1), cool (−2), and cold (−3). Users are able to provide participatory readings by using a cell phone application (IPhone and Android) or by a website. Figure 9.1 shows each of these input methods.

As user input could potentially be used for applications involving temperature actuation, security must be considered. In order to use the interfaces, authentication is required through the university central authentication system where only students and staff with university accounts can access the system. In addition, when occupants register for the service, they must submit a room request that is verified before approval. Before performing our studies, we obtained approval from our university Institutional Review Board. Best practices are followed with regards to use of data [OEC, eur].

As with any participatory sensing application, privacy is an issue that should be considered. Since thermal comfort level is not considered sensitive, most occupants feel comfortable providing this information. Occupants would have to report this information regardless if they were to request temperature change from management. Erring on the side of caution, we chose not to include any identifiers when storing user feedback in our database.

9.2 Crowd Driven Control

The first application we examine is a building temperature control system driven by data gathered by Thermovote. Our deployment covers one floor of an engineering building and one floor of a management and social sciences building. These spaces have a mix lab space, private offices, and open office spaces. Lab spaces are controlled using a Phoenix system [pho], which uses LonWorks as the network protocol. The remaining space is controlled using Automated Logic [Log], which is a BACnet based system. BACnet is a client-server protocol used for building automation and control. LonWorks is another similarly designed protocol used by Pheonix. Both systems are maintained and administrated using WebCtrl, which is a BMS made by Automated Logic. The floorspace is subdivided into multiple zones with each zone containing one or more rooms. Each zone contains multiple BACnet points, where each point either provides sensory information or point of control. For example, an air damper can be actuated by "writing" to the damper BACnet point. Temperature can be determined be "reading" the thermostat BACnet point of a zone. WebCtrl allows changes to be made to parameters such as temperature using a Simple Object Access Protocol (SOAP) interface.

The first major design issue is how to utilize user feedback. If a user provides a value of -3 (Cold), then how much should the temperature be adjusted? In order to determine an appropriate temperature change, we use the well known Fanger's PMV model (ASHRAE 55). Since we ask users directly for their thermal comfort level, we know the actual mean vote (AMV). Thus, we can set PMV equal to AMV and in order to correct the errors caused by estimates of the parameters. This error correction is discussed in greater depth in the following sections. By utilizing the AMV of an area, we also handle differences in preference. If the majority of the occupants in a particular area is warm, but only one is cold, then the system can make an adjustment that favors the majority. Since thermal preference can change over time, votes are tallied at regular intervals.

9.2.1 Baseline Control

Determining a set-points for temperatures is often an imprecise exercise. As previously mentioned, ASHRAE 55 specifies guidelines for creating comfortable temperatures through the PMV metric. In order to calculate the PMV, the following parameters are required:

- Metabolism (M)
- External work (W)
- Mean Radiant Temperature (T_{rad})
- Air temperature (T_{air})
- Relative Humidity (h)
- Partial water vapor pressure (P)
- Clothing coefficient (f_{cl})
- Air velocity (v_{air})

These parameters are used in the following to calculate PMV:

$$
\begin{aligned}
&P\hat{M}V(M, w, T_{rad}, T_{air}, h, C, v_{air}) = \\
&\quad (0.028 + 0.303e^{-0.036M})\{ \\
&\quad (M - W) - 3.05{-}3(5.733 - 6.99 \times 10^{-4}(M - W) - P) \\
&\quad -0.42((M - W) - 58.15) - 0.017M(5.867 - P_a) \\
&\quad -0.0014M(34 - t_a) \\
&\quad -3.96 \times 10^{-8} f_{cl}((T_{cl} + 273)^4 - (T_{rad} + 273)^4) \\
&\quad -f_{cl} \times (T_{cl} - T_{air})\}
\end{aligned}
\tag{9.1}
$$

where outer clothing temperature T_{cl} is

$$
\begin{aligned}
&T_{cl} = 35.7 - 0.028(M - W) - \\
&\quad I_{cl}(3.96 \times 10^{-8} f_{cl}((T_{cl} + 273)^4 - (T_{rad} + 273)^4) - \\
&\quad f_{cl} \times (T_{cl} - T_{air})\}
\end{aligned}
\tag{9.2}
$$

The main difficulty is that most of the parameters are not measured by a BMS. Typically, only air temperature and relative humidity are actually measured. While the BMS does sometimes have airflow through the vents, this does not accurately reflect the perceived airflow felt by the occupant unless the occupant is in the direct path of the vented air. Mean radiant temperature (MRT), the average weighted temperature of the surroundings, is also not commonly sensed. This value, affected by solar gain heating the surroundings, can be measured using a black globe thermometer. The other remaining parameters are estimated since factors such as metabolism cannot be practically sensed or measured. Table 9.1 shows the initial set-points for the building. Rooms with more sun exposure had heating set-points of 72 F^{o}. The remaining rooms had set-points of 74 F^{o}. Since many of the parameters are fixed estimates, the PMV error will vary depending

Parameter	High Solar Gain	Low Solar Gain
Metabolism	70 W/m^2	70 W/m^2
MRT	75 F^o	72 F^o
Relative Humidity	30%	30%
Clothing	1.0	1.0
Air Velocity	0.1 m/s	0.1 m/s

Table 9.1: Parameters used for initial PMV estimate and temperature set-points.

on conditions. For example, if the estimated parameters are based on a sunny day, then the PMV error will increase on cloudy days. Rooms that have more sun in the morning might have higher PMV during this period of the day, but have low PMV in the afternoon when the room receives no solar gain. The goal of our strategy is to continually correct for these dynamically changing environmental factors.

9.2.2 Learned Control Schedule

The learned control strategy uses historical data in order to correct temperatures using an open loop system. Users are asked to report their thermal comfort over the course of several days. This data is then aggregated to create a daily temperature schedule. The temperature at each point of the day is set based on the past user input. For example, a room may consistently be warm in the morning due to solar gain, which users can log during the “learning” period. These patterns are captured by aggregating votes at each point of the day.

Temperatures are corrected using user data and the PMV model. Fixed parameters are the main source of errors for PMV. Out of the PMV parameters, temperature is the most important factor affecting PMV that can be regulated by the BMS. Since we have actual data from the occupants, we have the actual mean vote (AMV) of the users. Thus, we can compare the PMV and AMV in order to correct for the errors. This can be achieved by offsetting the air temperature until PMV matches the AMV,

$$P\hat{M}V(M, w, T_{rad}, T_{air} + T_{offset}, h, C, v_{air}) - AMV = 0 \tag{9.3}$$

In this equation, T_{offset} is a temperature offset that will correct the initial PMV estimate to match AMV provided by occupants. This method allows us to translate a AMV such as “Cold” to an actual temperature change based on the current conditions.

The learned algorithm is shown in Algorithm 6. The day is subdivided into 10 minute windows of time. For each window, the historical user votes for the window of time is examined to find the AMV. For example, if we gathered a week of data and are examining the period 14:00-14:10, then we calculate the AMV by aggregating the votes of this 14:00-14:10 period across all 7 days. The AMV is then used to calculate the temperature change T_{offset}. However, we do not change the set-point if the difference between the air temperature and the set-point is less than a specific threshold. In this case, the system is adjusting the room to the specified set-point. The threshold is determined by examining the amplitude of the temperature change caused by the PID controller. Only one vote is allowed per user during each 10 minute period, which is done to prevent bias.

Since most modern thermostats can be programmed with daily temperature schedules, this method can be applied to the majority of current systems and does not require a BMS. Offsets can be pre-computed and programmed as a schedule. However, there are drawbacks to this approach. In particular, the offsets are based on the time of collection. Data collected in the summer may not apply during the winter. Seasonal error can be reduced by collecting data for each season but will not be able to adapt to heat waves or cold spells.

Algorithm 6 Learned Control Schedule

```
$\hat{PMV} \leftarrow$ See Equation 9.1
$AMV(i, j) \leftarrow$ Actual Mean Vote from previous data from time $i$ to $j$
$T_{setpoint} \leftarrow$ Current temperature set-point
$t_i \leftarrow$ Time at instance i
$thresh \leftarrow$ Threshold to consider $T_{air} \approx T_{setpoint}$

for Every $n$ minutes do
  $t_{current} \leftarrow$ Current time

  Solve for $T_{offset}$:
  $\hat{PMV}(M, w, T_{rad}, T_{air} + T_{offset}, h, C, v_{air}) - AMV(t_c, t_{c+n}) = 0$

  if $|T_{setpoint} - T_{air}| < thresh$ then
    $T_{setpoint} = T_{air} + T_{offset}$
  end if
end for
```

9.2.3 Real-Time Control

An increasing number of BMS's allow for real-time control. With real time control we can apply a similar strategy to the learned control schedule, except we can apply the corrections based on real-time user readings. Algorithm 7 shows the control algorithm. The strategy is similar to Algorithm 6. However, instead of aggregating results from previous data, data from the past n minutes (10 minutes) is used instead. Again, to prevent voting bias, we only count the most recent vote by a user for each 10 minute period.

The main advantage of this approach is that it is highly adaptable. In particular, it can account for sudden changes such as someone opening a window, changes in occupant thermal preference, or extreme weather. In Section 9.4, we will also see that it is robust against the variance of user feedback. Though real-time temperature control is currently less common, increasing availably of real-time temperature control will eventually allow widespread application of this strategy.

9.3 User Studies

For our study, we recruited a total of 135 participants in two different buildings to provide feedback, covering approximately 85% of the total occupancy of the floor in each building. Table 9.2 shows the demographics of the users. We held information sessions and described our study to people occupying the target floors. For our initial studies, we conducted three different studies over a period of 5 weeks in Lab 1 and Office areas (31 participants). After these studies, we continued to run the real-time studies in these areas and added an additional Lab 2 area. Figure 9.4 shows the zones of the office space. One of the long term study has now run for total of 25 months.

Study 1: Baseline Evaluation: During the first week baseline period, occu-

Algorithm 7 Real-Time Control Schedule

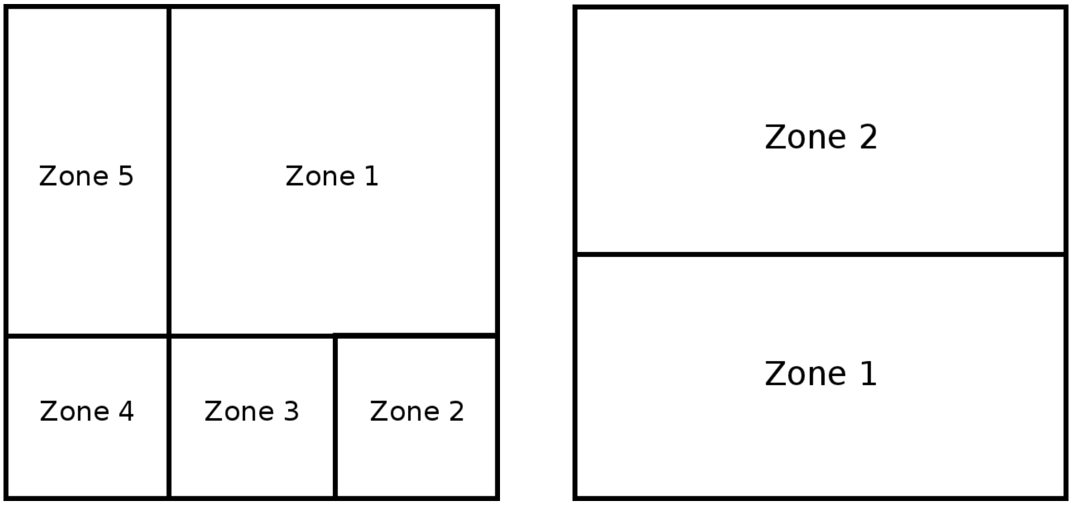
```
$\hat{PMV}$ ← See Equation 9.1
$AMV(i, j)$ ← Actual Mean Vote from current data from time $i$ to $j$
$T_{setpoint}$ ← Current temperature set-point
$t_i$ ← Time at instance i
$thresh$ ← Threshold to consider $T_{air} \approx T_{setpoint}$

for Every $n$ minutes do
    $t_c$ ← Current time $c$

    Solve for $T_{offset}$:
    $P\hat{M}V(M, w, T_{rad}, T_{air} + T_{offset}, h, C, v_{air}) - AMV(t_c, t_{c-n}) = 0$

    if $|T_{setpoint} - T_{air}| < thresh$ then
        $T_{setpoint} = T_{air} + T_{offset}$
    end if
end for
```

	Male	Female
Gender	67%	33%

	21-29	30-39	40-49	50 or older	Declined
Age	48%	36%	6%	3%	6%

	Faculty	Staff	Student
User Type	12%	37%	51%

Table 9.2: Demographics of the users.



(a) Office

(b) Lab 1 and 2

Figure 9.4: Layout of offices and labs.

pants logged their thermal preference to determine how well the static temperature schedule performed. In addition to establishing baseline performance, this data was used to create a learned schedule for the second study. A survey was also given to gather information such as overall satisfaction, area conditions, and past interactions with the Facilities Department.

Study 2: Learned Control Schedule: This one week study uses the data gathered from the baseline study to learn the temperature offsets to correct PMV estimates using the technique described above. To determine the effectiveness of the learned schedule, occupant were asked to continue to log their comfort level. We sent another survey asking occupants about their overall satisfaction with the new temperature schedule.

Study 3: Real-Time Control: This two week study tests the real-time control strategy described in the previous section. A survey is sent to gather user feedback regarding this strategy.

Long Term Studies: Real-Time Control: After the initial 3 studies, we conducted several long term studies to examine how users interact with the real-

	Within 15 feet
Window	59%
Exterior Wall	72%
Window and Wall	56%

Table 9.3: Proximity to window or exterior wall.

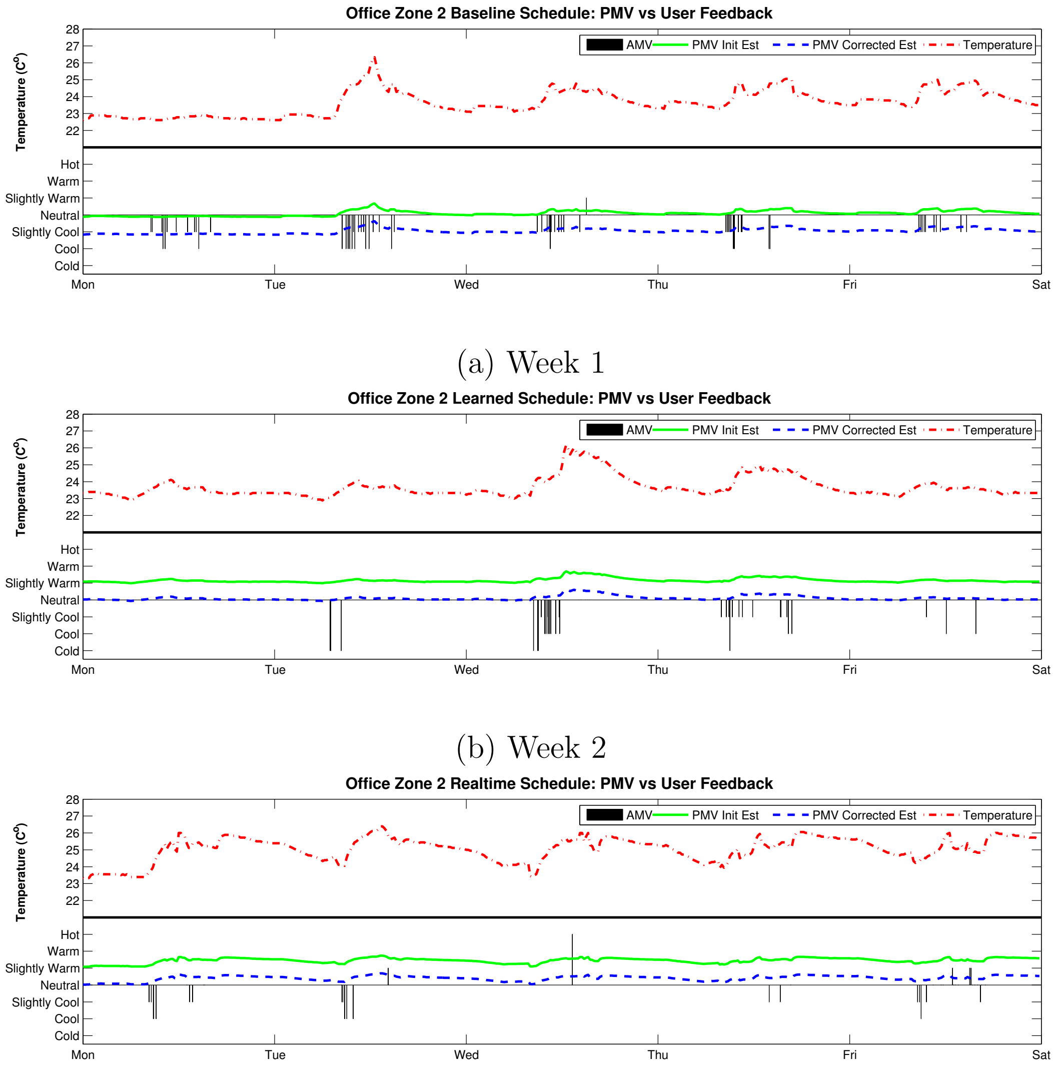
time system. Our longest ongoing study is currently for a 25 month period. We also have an additional study that has currently run for 15 months. Surveys were given at the beginning and after 5 months to determine the prior performance of the baseline system and the user perception of the system.

9.4 User Studies

For our study, we recruited a total of 135 participants in two different buildings to provide feedback, covering approximately 85% of the total occupancy of the floor in each building. Table 9.2 shows the demographics of the users. We held information sessions and described our study to people occupying the target floors. The buildings were constructed in 2006 and 2012 and are both LEED Gold certified. For our initial studies, we conducted three different studies over a period of 5 weeks in Lab 1 and Office areas (31 participants). After these studies, we continued to run the real-time studies in these areas and added an additional Lab 2 area. Figure 9.4 shows the zones of the office space. One of the long term study has now run for total of 22 months.

9.4.1 Surroundings

A preliminary survey was given to determine the surrounding environment of occupants. Table 9.3 shows how many occupants are within 15 feet of a window or external wall, which can potentially affect comfort level. 59% of occupants



(a) Week 1

(b) Week 2

(c) Week 3

Figure 9.5: The top figure shows the initial estimate PMV estimate for this area, which differs greatly from the user feedback. Based on this initial PMV estimate and user feedback, we can correct the PMV estimate. The middle figure shows when we condition according to the corrected PMV estimate. Though people still indicate they are cold, the frequency of the vote is reduced. The bottom figure shows the system running in real time. The frequency of the vote is reduced dramatically indicating that occupants are comfortable. The corrected PMV is actually higher than 0, indicating that our original correction produces temperatures that are slightly cool.

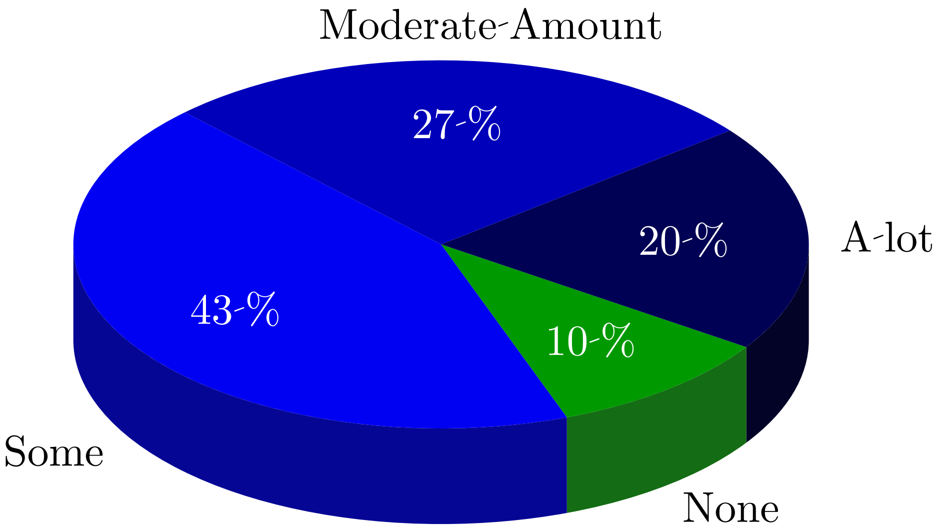


Figure 9.6: Airflow experienced by users.

were within 15 feet of a window and 72% were within 15 feet of an external wall. We also asked occupants to report the perceived amount of air felt from the VAV. Figure 9.6 shows most people felt some degree of airflow (84%); only 9% did not feel any air movement.

9.4.2 Thermal Comfort: Studies 1-3

In Figure 9.7, we see the vote results over a 5 week period for two representative zones. We first examine the baseline performance (Study 1). The initial PMV estimate shown in Figure 9.5(a) is calculated using the initial parameters shown in Table 9.1 (high solar gain). This estimate is neutral (PMV 0) most of the time with some variation caused by the natural air temperature changes in the room. However, when we compare the PMV estimate with the actual votes, we see most occupants felt Cool to Colder during this period. This is also confirmed from the initial survey. Table 9.4(a) shows that 76% of the occupant in the area either cold, cool, or slightly cool. 0% of the respondents indicted they were satisfied or even slightly satisfied with the initial conditions; 75% indicated they were not satisfied with the initial conditions. Based on the votes and surveys, a positive

(a) Thermal Comfort

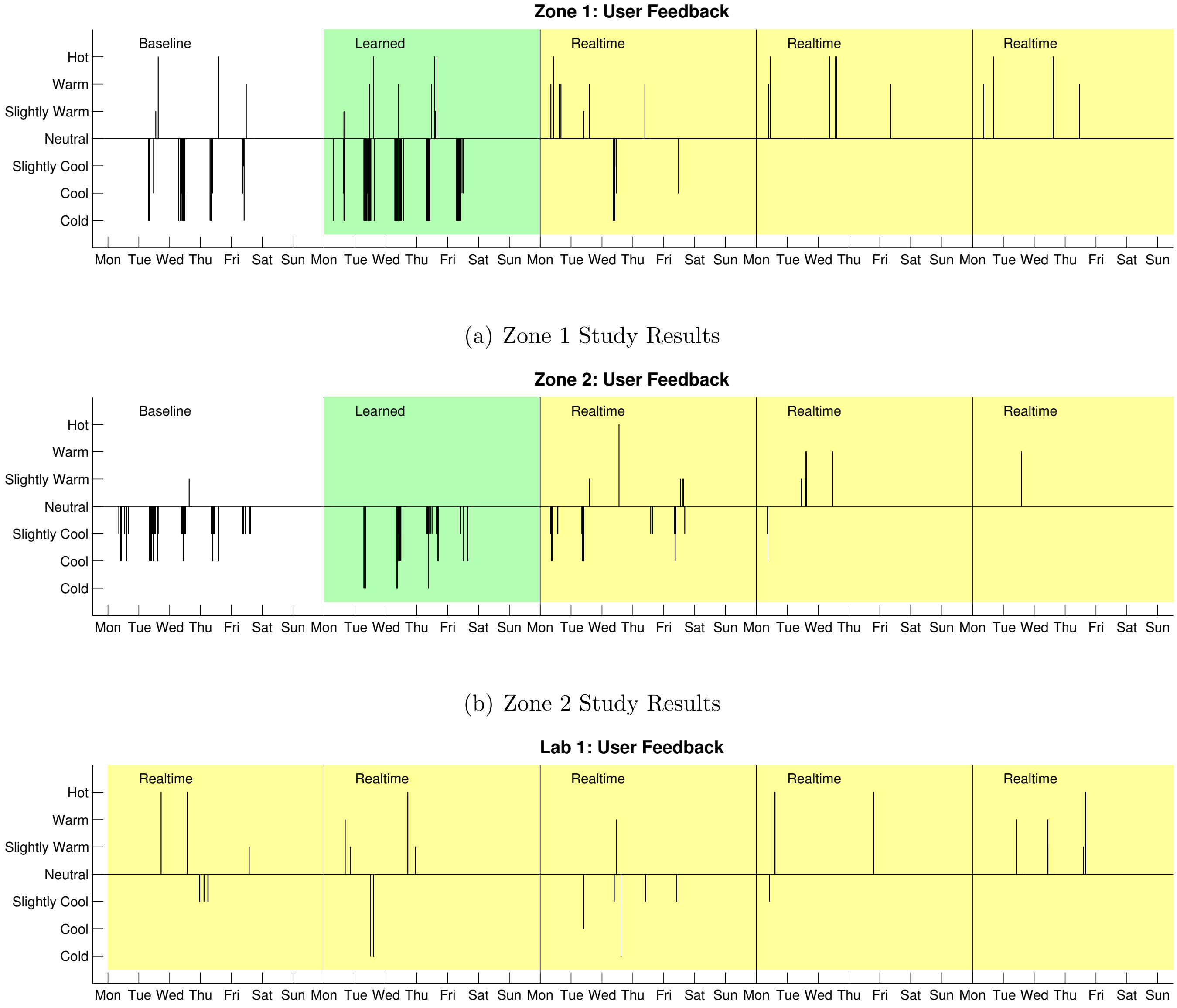
	Baseline	Learned	Real-time
Cold	29%	27%	0%
Cool	29%	9%	0%
Slightly Cool	18%	9%	22%
Neutral	0%	19%	67%
Slightly Warm	12%	27%	11%
Warm	12%	9%	0%
Hot	0%	0%	0%

(b) Satisfaction

	Baseline	Learned	Real-time
Dissatisfied	33%	8%	0%
Somewhat Dissatisfied	42%	17%	0%
Neutral	0%	8%	0%
Somewhat Satisfied	8%	50%	77%
Satisfied	17%	17%	23%

Table 9.4: Survey results for studies 1, 2, and 3.

temperature offset is required in order to correct the PMV estimate. The AMV is calculated by averaging the votes over 10 minute windows where the total we average over is the number of unique occupants who have voted for the current day. Occupants that do not vote during a window are assumed to have vote of 0 (neutral). We found occupants typically vote when they are uncomfortable, not when they are comfortable. The calculated AMV is then used to correct the PMV estimate. The corrected estimate is shown in Figure 9.5(a), which shows corrected PMV estimate closer to the actual votes.



(a) Zone 1 Study Results

(b) Zone 2 Study Results

(c) Lab 1 Study Results

Figure 9.7: Study results over the course of 5 weeks.

The learned strategy (Study 2) that was applied during the second week showed improved thermal comfort. Rooms showed AMV values closer to 0, and the rate of voting decreased, showing that temperatures had improved (see Figure 9.5(b)). Survey results (Table 9.4) show 19% of occupants were comfortable compared with the inital 0%. Only 45% felt slightly cool or cold with the learned strategy vs the 76% felt slightly cool to cold during baseline. We also found that a larger number of occupants were warmer (36%). This indicates that our corrected PMV still contains some error for certain areas. In particular, it shows corrections were too large in certain areas and too little for others. However, overall occupants were more satisfied with the learned temperature schedule (Table 9.4). 67% were satisfied or somewhat satisfied using the learned strategy; the corresponding results for baseline was only 25%.

Study 3 showed the real-time strategy performed significantly better than both baseline and learned strategies. Figure 9.7(c) shows the total number of votes decreasing for several areas. As occupants adjust the temperature, less votes are required to maintain comfortable conditions. The survey showed majority of occupants felt comfortable (67%); only 22% felt slightly cool and 11% felt slightly warm. Overall we show that 77% were satisfied and 23% were somewhat satisfied.

9.4.3 Thermal Comfort: Long Term Studies

We expanded our deployment area and launched two long term studies; at the writing of this dissertation, a real-time system has run 25 months in the Office, Lab 1, and Lab 2 areas and another real-time system has run 15 months using an additional set of 5 offices (GradOff) occupied by graduate students. The GradOff areas are each zoned separately. Baseline surveys of thermal comfort show that initially 0% found the rooms to have neutral temperatures. Table 9.5 shows the survey results for occupant thermal comfort levels. 62% of the occupants found the space they occupied to slightly warm to hot. 38% found their space to slightly

(a) Thermal Comfort

	Baseline	Real-time
Cold	28%	0%
Cool	25%	9%
Slightly Cool	9%	10%
Neutral	0%	65%
Slightly Warm	16%	12%
Warm	16%	4%
Hot	6%	0%

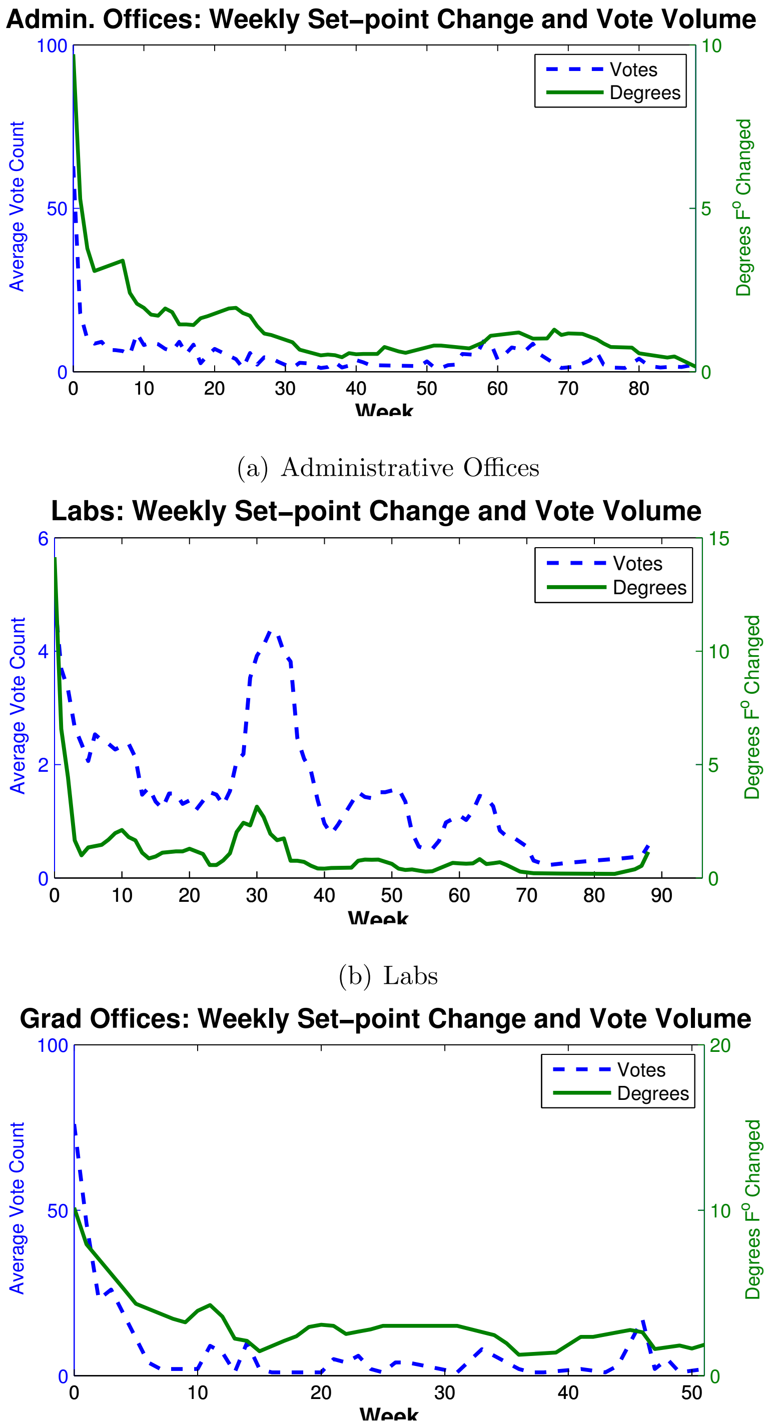
(b) Satisfaction

	Baseline	Real-time
Dissatisfied	44%	0%
Somewhat Dissatisfied	37%	7%
Neutral	0%	13%
Somewhat Satisfied	13%	27%
Satisfied	6%	53%

Table 9.5: Long term comfort and satisfaction.

cool to cold. Table 9.5 shows the baseline and long term satisfaction from surveys. Baseline performance shows that only 19% were satisfied or somewhat satisfied with initial conditions. After using the real-time system over a long period, 80% were satisfied or somewhat satisfied; only 7% were dissatisfied. One interesting finding is that despite only having 65% of occupants indicate neutral comfort, 80% of occupant were satisfied. This shows that absolute thermal comfort is not necessary to achieve satisfaction.

In addition to thermal comfort, there were several issues that we wished to examine in this study. In particular, we wanted to examine the user participation over time. Figure 9.8(a) shows the change in vote volume and total temperature change over time for the different room types. Initially, the vote volume is high. This is likely due to the novelty of the system for users. Consequently, the total temperature change is also high. After the first few weeks, as the room temperature converged to a comfortable range, we see the vote volume decrease similar to what was seen in Study 3. From week 15 to 30, the temperature change is between 1.0-1.9 F^{o}. From weeks 30 onward we see that the temperature change only between 0.1-1.0 F^{o}. We see similar results for the graduate offices (Figure 9.8(c)); we see a high initial vote volume the first few weeks that eventually tapers in the following weeks. For these areas we see changes between 1.5-3.0 F^{o}. Even though the vote volume is similar to the administrative area, the temperature changes were higher for the graduate area as compared the administrative area. This was due to the lower bias we saw with the votes of the graduate students, which caused less votes to "cancel" out. We saw a similar situation with the Lab areas where we again see higher temperature changes relative to the number of votes. However, for the Lab areas (Figure 9.8(b)) we see a slight variation with regards to the long term voting pattern. In this case, we see the familiar initial experimentation phase in the beginning followed by a decrease number of votes, followed by another increased spike around week 32. This was due to a new group



(a) Administrative Offices

(b) Labs

(c) Graduate Offices

Figure 9.8: The weekly vote average and temperature change for different areas.

of graduate students entering the study. Again in the weeks following, we see the number of votes tapers off.

During the seasonal changes, weeks 10-30 and 60-70, the degrees changed clearly increased for the Lab and Administrative Offices (Figure 9.8(a),(b)). Interestingly the Graduate Offices (Figure 9.8(c)) did not show significant variation of degrees changed. When we examined these rooms, we realized that these spaces generally had less window area and exterior wall space than the Labs and Administrative Offices. The building housing the Graduate Offices also is a newer building with slightly better insulation and this factor is likely the cause of the reduced correlation between weather and voting patterns for these particular areas.

We attribute lower vote volume to more comfortable temperatures; votes are usually supplied when users are uncomfortable. Another interpretation of the results is that occupants are losing interest and not voting. To test this hypothesis, we performed a simple interrupted time series [DC02] by turning off the Thermovote in one zone without prior notice to the occupants. An office zone, which experiences similar conditions, serves as a baseline. Stopping “treatment” leads to a dramatic increase in the voting volume immediately following the deactivation. Week 5 and 6 of this test (Figure 9.9) shows the period when Thermovote was deactivated. In the month prior, weeks 1-4, the average vote volume was between 1 to 2 votes per day. During weeks 5 and 6, the average vote volume per day increased to 5-7. After re-starting Thermovote, the vote volume again dropped to previous levels. The baseline showed 1-2 votes per day for the same period of time confirming the deactivation of Thermovote was likely the main cause for the change in vote volume. These results show that occupants still actively use the system and rely on it for temperature maintenance; it is not a novelty. The findings also show that system is effective at correcting thermal estimates and only a few votes per day are required to maintain proper room conditions.

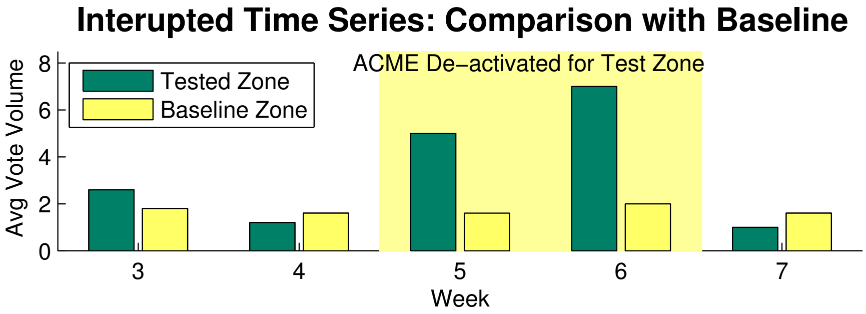


Figure 9.9: Effect of activating and deactivating Thermovote.

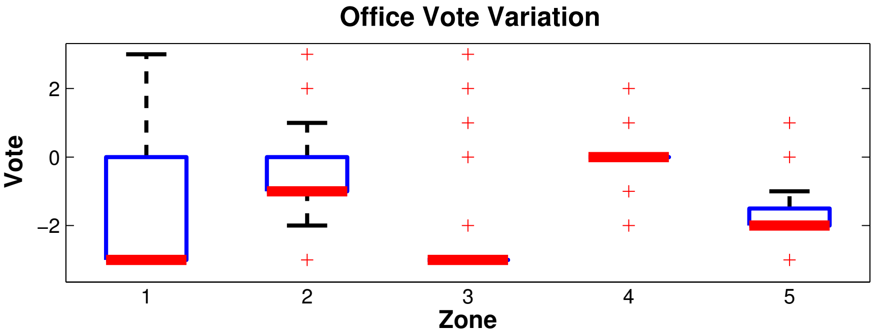
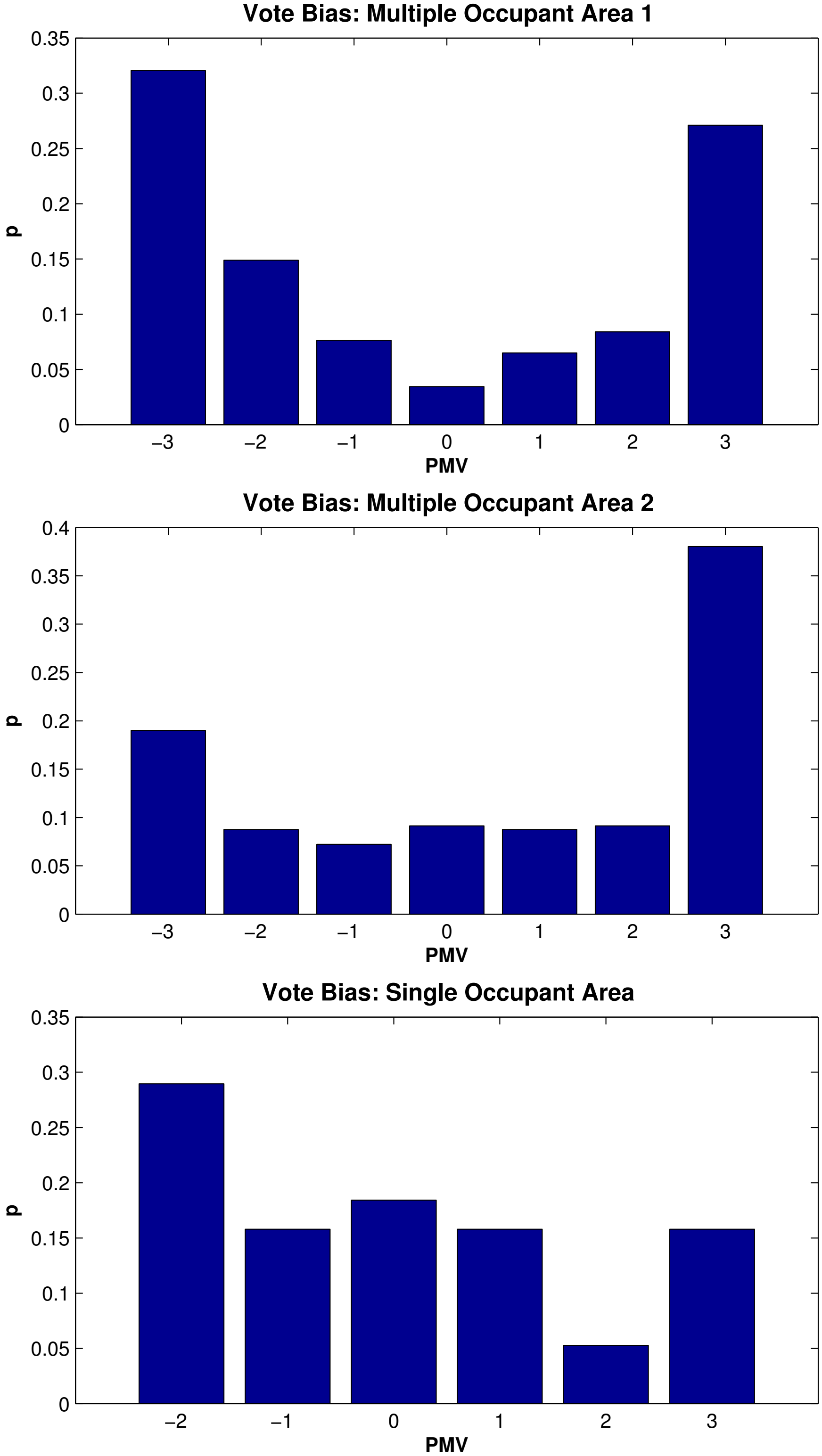


Figure 9.10: Variation of votes for zones with multiple occupants. The dark thick line is median.

9.4.4 Vote Consistency

Another issue of interest is the consistency of the perceived thermal comfort. Since user opinion is the definitive authority, an individual's "sensed readings" can be considered 100% accurate. However, an individual's perceived thermal comfort level may not match the overall perceived comfort level of the group since there are differences among individual preference. This variance is shown in Figure 9.10. Zones 3 and 4 show votes that are consistent among occupants with only a few outliers. The convergence of the votes suggests that people tend to have similar thermal preferences. People are more likely to agree that a particular temperature is comfortable. However, people are less likely to agree on the severity of uncomfortable temperatures. While one person may feel that a room is cool, another person may feel the room is cold. Though we use the same temperature correction method for both real-time and learned schedules, we see conditioning performance of the real-time system is superior. This superior performance is most likely due to the real-time feedback returned to the user. Users can re-correct the temperatures until they feel comfortable, functioning as a human driven proportional integral derivative controller.

Zone 1, however, shows a larger variation, where most occupants voted “Cold” (-3) but a significant number of “Hot” votes were also recorded during. While we may expect users to disagree on severity (Cold vs slightly cool), it is puzzling to see such a large divergence in opinion. The most likely explanation is that the minority is aware of the majority’s preference. In this case, the minority may attempt to offset the majority’s vote by voting for the extreme opposite. This variation is also seen in Office Zones 2 and 5 but to a lesser degree. Figure 9.11 shows the votes in two different areas where multiple people occupy the space along with a space only occupied by a single person. Area 1 occupied by multiple people shows a clear pattern of bias caused by competition. There is somewhat a division of preference in this particular space with slight majority of people feeling the space is cold to slightly cool. Area 2 occupied by multiple people shows a more definite consensus with more people reporting Hot (38%); we also see a small percentage of people voting Cold (18%) in order to bias the temperature in their favor. The remaining points from -2 to 2 are very uniform and are between 8%-10%. If we examine the single zone space we see a very even number of percentages for PMV -1, 0, and 3 (15%-18%). The user only found the space warm about 5% of the time. There was a higher percentage of -2 in this case since the room was initially too cold and required extra votes to adjust the temperature.



Vote Bias: Multiple Occupant Area 1

Vote Bias: Multiple Occupant Area 2

Vote Bias: Single Occupant Area

Figure 9.11: Vote bias for areas occupied by multiple/single occupants.

9.4.5 Maintenance and Management

In addition to thermal comfort, another important factor to consider is system management. For building management, reducing the number of complaints is of critical importance. In our surveys, we also included questions regarding users' experience when interacting with facilities management.

Users only contacted us 6 times during the long term real-time study period. One zone have extreme temperatures. This particular zone had three small offices attached to a single variable air volume (VAV) unit, where each room receives a portion of air from the shared VAV. The amount of air each room receives is manually controlled by adjusting a series of vents. In this particular case, one room received 200CFM while the remaining rooms only received 50CFM each. We contact facilities and resolved the issue through an air balance of the areas. There were 3 instances where network/power outages caused our vote system to go off-line. The remaining contacts were simple questions about the system. Survey results showed 100% of occupants were either satisfied or somewhat satisfied with our resolution.

As shown in Table 9.6, before the implementation of Thermovote, 56% of occupants contacted facilities because of thermal comfort issues. 31% of these occupants were satisfied or somewhat satisfied. If 81% of occupants were dissatisfied with thermal comfort (Table 9.5) and only 56% of occupants contacted facilities, then approximately 25% choose not to contact facilities. Given that most occupants were not satisfied with the results, it is possible the 25% occupants who did not contact facilities perceive contacting facilities as ineffective. While this may seem somewhat shocking, it is expected since even if facilities responds to each complaint instantly, they do not incorporate all previous user feedback. Thus, it is unlikely the temperature selected satisfies everyone.

We also found the system to detect other faults within the system. In one in-

(a) Year Contact

Did Not Contact	37%
Unable to Contact	6%
Did Contact	56%

(b) Resolution

Dissatisfied	38%
Somewhat Dissatisfied	25%
Neutral	6%
Somewhat Satisfied	25%
Satisfied	6%

Table 9.6: User interaction with facilities management before Thermovote.

stance, we detected a sudden increase of warm votes within the Lab 1 spaces. Though votes were being recorded, temperatures were not being accepted by the BMS. Upon further inspection, we found the VAV was locked in a default state, which was preventing changes to the system. Since the default state caused warmer conditions, user feedback alerted us of this situation. This occurred several times over the months. To prevent loss of service, we configured our system to email us when the VAV was in a locked state so we could reset the VAV. Without the user feedback, detecting these types of problem is difficult and it is likely this default state would have persisted for several months before being discovered. Thermovote has also been used to examine the effect certain control changes had on occupants. For instance, facilities was adjusting the set-point control logic of the air handler unit. In particular, adjustments were being made to the economizer of the air-handler to increase efficiency. One concern was that temperature changes could affect the air flow within the spaces causing colder temperatures. Thermovote was used in this case to verify that proper conditioning was still taking place.

9.4.6 Energy Consumption

In this section we examine how the strategies affect energy consumption. To estimate the energy consumption, we analyze the heat transfer through the variable air volume (VAV) units. Each VAV contains a heat exchanger that delivers conditioned water. Air is passed over the heat exchanger, conditioning the air for the room. Air for the driven through the VAV units using the main air handler unit on the roof. To measure the energy from the heat transfer, we use the following: $Q = mC_{air}(T_{in} - T_{out})$ where Q is energy transfered from the coil to the air, m is the total air mass passing through the coil, C_{air} is the specific heat of air, T_{in} is air temperature of the incoming air, T_{out} is the temperature of the outgoing air after passing through the coil. The air mass is estimated by examining airflow to find the total volume of air and then using air density.

For the learned strategy, we see a 2% increase in energy when compared to baseline. This shows the overall change of the room offsets did not significantly change the amount of energy used. Though additional heating was required for the majority of the areas, there were several areas receiving a large amount of solar gain that actually required a small amount of cooling. Since the outside air temperature was sufficient for cooling needs, these areas did not contribute to the energy consumption. The real-time system showed 10.1% savings over the baseline strategy, showing that in certain situations a real-time system can potentially save energy. The baseline used for this calculation was found using similar days from historical data. This is to account for weather, which can affect energy consumption. The real-time week had an average temperature of 54.4 F^{o} and standard deviation of 1.9 F^{o}. The corresponding baseline week had an average temperature of 55F^{o} and standard deviation of 1.5 F^{o}. On average office zone temperatures increase 2.1 F^{o}, increasing energy consumption. However, temperatures in the labs decrease 3.3 F^{o}, accounting for the net energy savings.

Next we analyze the energy consumption for the 25 months study. Overall we found a 16% increase in energy consumption for the 25 months. We again pick baseline data to match the weather conditions during the real-time deployment. Offices 1 and 2 showed increases of 9% and 21% respectively while Office 3 showed a change of -14%. This indicates the effect on energy consumption depends on the situation. For example, for a room with a temperature that is too cold in the summer, Thermovote will likely decrease energy. Similarly, for a room that is too warm in the summer, Thermovote will likely increase energy. This is further supported comparing the results of [EC] with our work; the authors found an overall decrease in energy for the period of time their system was evaluated, while our current deployment found an overall increase in energy.

An important point to make is that energy savings should not come at the expense of occupant comfort. Previous to Thermovote the building as a whole consumed less energy. However, this energy saving came at the cost of thermal comfort, which is confirmed by the user feedback. This discomfort also has a monetary cost in terms of lost office productivity. In [Hed04], typing errors dropped 44% and output increased 150% by increasing temperatures from 68F^o to 77F^o. Thus, before energy saving measures are considered, one should also consider the thermal comfort of the users.

9.5 Thermal Comfort Models

In the remaining sections, we discuss how the Thermovote system is an enabling technology that allows the collection of thermal comfort data from users and enables the construction of models of thermal comfort based on empirical data. These models can be integrated into energy simulations and model driven control schemes, such as model predictive control (MPC) [KB11] to actuate buildings in a more efficient manner.

PMV is currently the most commonly used model for thermal comfort. As previously mentioned, many of the parameters of this model must be estimated. This leads to a very deterministic value of comfort. In reality, there are typically a range of comfort values for a given room temperature. Thus, it can be argued that

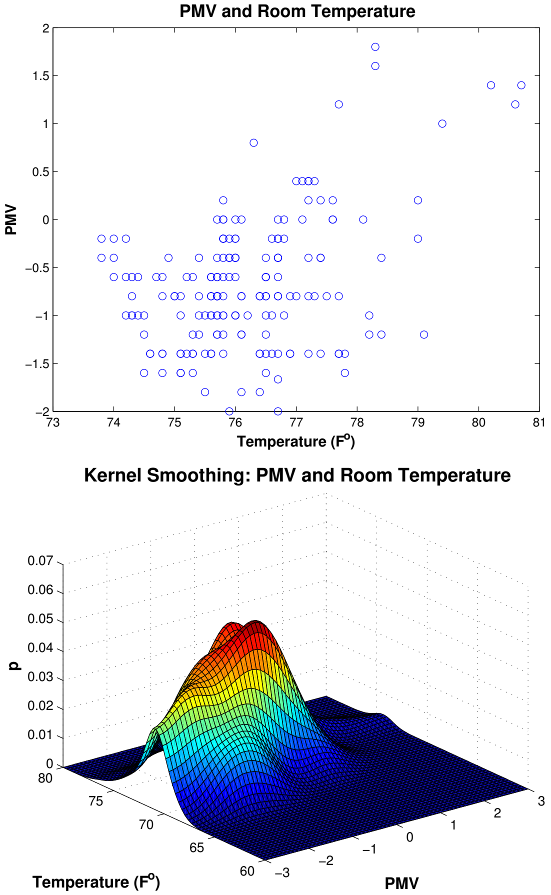


Figure 9.12: Comparison of the original data and KDE.

	Room Temp.	Outside Temp.	Humidity	Wind
r	0.801	0.056	0.017	-0.145

Table 9.7: Correlation coefficient of thermal comfort with several parameters.

it is more practical and realistic to model comfort levels based on a distribution. In the following sections we develop and examine two different comfort models using data collected from Thermovote.

9.5.1 Model Development

We start by first examining the parameters that can easily be obtained from the building automation system or weather data. Table 9.7 shows the list of parameters examined and the correlation coefficient of these parameters when compared with thermal comfort.

As expected, room temperature had the highest correlation ($r = 0.801$). Given these findings, we use room temperature to develop a model using kernel density estimation (KDE). In the following sections, we show how these methods can be used to construct a probability density function (PDF) for occupant comfort given room temperatures.

KDE is a non-parametric method of estimating a PDF. This method estimates the density at a point by taking a weighted average of nearby samples where the weighting is determined by some kernel function. Let $X = (X_1, ..., X_n)$ be n independent and identially distributed samples of an unknown density $\hat{f}_h$. The kernel density estimator of $\hat{f}_h$ is

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\Big(\frac{x - X_i}{h}\Big)$$

where $h > 0$ is the smoothing bandwidth, $K(\bullet)$ is a symmetric kernel function, and $\hat{f}_h(x)$ integrates to 1. In our case, we define K to be a normal distribution. We set the bandwidth partly by trying statistical rules and partly by eye. Figure 9.12 shows the original 194 data points and the fitted PDF.

9.5.2 EnergyPlus Simulated AMV

In this section, we show how the models developed can be integrated with EnergyPlus [Epl] (EPlus) to explore different comfort based strategies and how these strategies affect energy consumption. EPlus is an energy management simulator that is able to account for a variety of parameters such as building geometry, weather, and occupancy to estimate energy consumption and examine how the building behaves. Natively, EPlus does not support customized models for thermal comfort. In order to integrate a thermal comfort model, we utilize MLE+ [Ngh], which is a package that is able to interact with EPlus and change parameters during simulations. This package allows halts at each simulation iteration and can be used to examine the state of the building. For our application, we will use the thermal model developed in the previous section to simulate votes from occupants. The model will give us distribution of occupant votes based on the temperature. Algorithm 8 is used for our energy simulations.

At each simulation iteration, we first examine each occupant to determine if a vote should be recorded. This is based on an exponential distribution or the duration between each vote fit from the original data. The number of occupants is based on an occupancy schedule. Based on all the votes cast, we calculate the estimated actual mean vote $A\hat{M}V$ by averaging all the votes during this period. We then calculate the appropriate T_{offset} in a similar manner as the previously live deployment strategy. However, rather than adjusting PMV to be 0, we calculate T_{offset} such that the estimated PMV falls within acceptable range of thermal comfort determined by an AMV set-point.

Algorithm 8 Learned Control Schedule

```
$P\hat{M}V \leftarrow$ See Equation 9.1
$\hat{AV}(T_{air}) \leftarrow$ Modeled Actual Vote given $T_{air}$
$T_{setpoint} \leftarrow$ Current temperature set-point
$AMV_{setpoint} \leftarrow$ AMV setpoint for acceptable comfort range
for Each Simulation Iteration do
    for Each occupant do
        if occupant scheduled to vote then
            $votes[occupant] = \hat{AV}(T_{air})$
        end if
    end for
    $A\hat{M}V \leftarrow mean(votes)$
    Solve for $T_{offset}$ such that:
    $-AMV_{setpoint} \leq P\hat{M}V(M, w, T_{rad}, T_{air} + T_{offset}, h, C, v_{air}) - A\hat{M}V) \leq AMV_{setpoint}$
    $T_{setpoint} = T_{air} + T_{offset}$
end for
```

We tested two different comfort based strategies in energy simulations. Since the number of samples is relatively small, we use KDE to model $\hat{AMV}$. Our target building is a multi-room multi-story building. The first strategy we tested was the same strategy utilized in the live deployment where $AMV_{setpoint} = 0$. The second strategy we tried was setting to $AMV_{setpoint} = 0.7$, which is the maximum value allowed by ASHRAE 55 standards. The same weekly occupancy schedule was used for each strategy. Figure 9.13 shows energy results of each of these studies as compared with a baseline strategy where the set-point band was 70-74F^o, which is often the starting set-points for many buildings. Overall, the optimal comfort $AMV_{setpoint} = 0$ strategy showed 7% increase overall annually. The winter months showed the highest increases (9%-17% for Dec-Feb) and the lowest increases during the warmer months (2%-4% for May to Apr). As expected, when we relaxed the comfort set-point to 0.7, we saw a reduction in energy usage (13% annually). The largest savings were during the colder months (9%-13% Nov-Feb). The smallest savings occurred during the warmest months (5%-6% for Jul-Aug). These results confirm that the energy costs are greater for heating; correspondingly, savings are thus also greater during heating. Figure 9.14 shows the vote distribution for both strategies. When the AMV set-point is 0, we can see the peak of the distribution is 0. When the AMV set-point is 0.7, we see one peak at 0.7 and another at -0.7. This is expected as the offsets at each iteration are just enough to move the temperature to within the acceptable range. During warmer weather, the offsets will push the AMV to just below 0.7. Similarly, during cold weather, the offsets will push it just above -0.7. We can also see the peak around -0.7 is higher than the peak around 0.7. This indicates that there are more instances of cold conditions than warm conditions.

These results show how data from Thermovote can be used to build much more realistic models of thermal comfort and how the models can be integrated into energy simulations to explore different design decisions that influence thermal

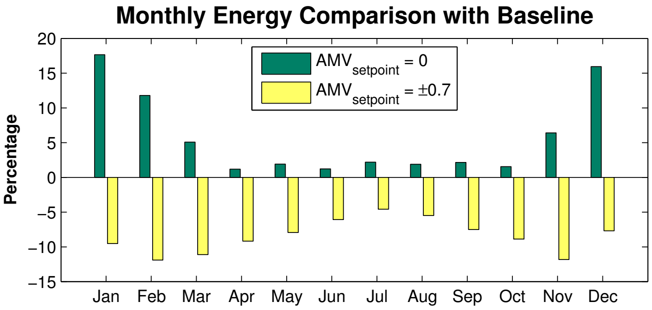


Figure 9.13: Energy above baseline.

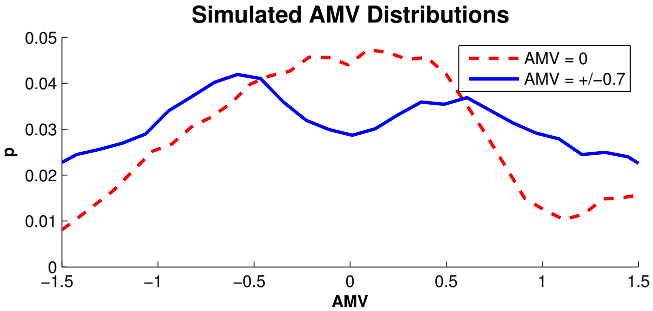


Figure 9.14: Simulated AMV Distributions.

comfort and vice versa. These results demonstrate just one simple application that a thermal preference simulator can provide; it can examine the interactions of thermal comfort with other parameters or be used to develop comfort based control strategies.

9.5.3 EnergyPlus Simulated Input Bias

In Section 9.4.4, we saw that competing thermal preferences could cause bias in the data; areas containing multiple people tended to have votes at the extreme ends of the scale in order to bias the temperatures in their favor. For our live deployments, our systems utilized the 7 point PMV scale, allowing for this introduction of bias data. One possible strategy, would be to use a 3 point scale (Cold, Neutral, Hot) in order to reduce the magnitude of the bias. In this section, we will use a building simulator along with a user thermal comfort model developed with past

Point Scale	Convergence Time	Std. Dev.
3 points	17.7 days	0.4765
5 points	11.6 days	0.4694
7 points (Simulated)	9.5 days	0.4417
7 points (Actual Avg.)	10.0 days	0.3233

Table 9.8: Point scale results, multiple occupants.

Thermovote data to test both the behavior of the system and how the users are likely to interact if we were to change the voting scale.

While it may seem preferable to test different point scales in a live deployment, the main problem is that it is impossible to replicate the same conditions even if experiments are run in parallel; too many variables exist that can influence both the user input and the system.

9.5.4 Voting Scale Comparison

Using our thermal comfort model, we are able to examine how changing the voting scale affects how the control system performs. In particular, we would like to examine how well lower point scales are able to adjust for different bias. For our simulations, we tested a 3 point scale (-1 to 1), a 5 point scale (-2 to 2), and a 7 point scale (-3 to 3). Scales above 7 points are not considered since it has been shown that beyond this resolution, people are unable to reliable distinguish the difference between each point within the scale [O70]. We utilize the same temperature correction procedure described in Section 9.2.2 but restrict the maximum correction depending on the scale used. For the 3 point scale, we solve for offset corrections setting PMV=-1,0,1. Similarly, for the 5 point scale, we solve for offset corrections setting PMV=-2,-1,0,1,2. For our tested control strategy, we choose to reduce the magnitudes of the temperature offset corrections and thereby reduce bias. Another control strategy is assuming the largest temperature corrections

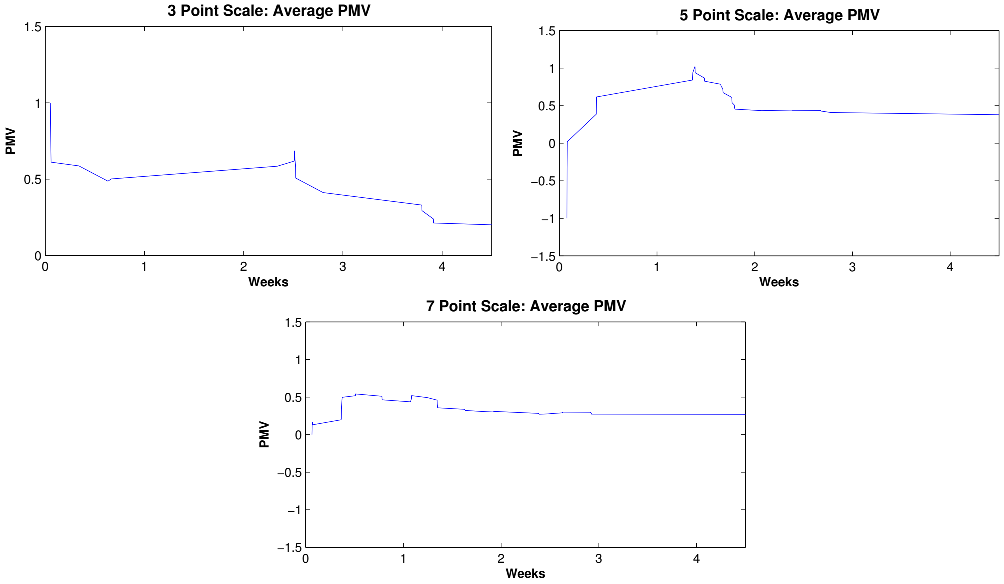


Figure 9.15: The convergence of the different scales for rooms with multiple occupants.

and use "0" as an input to stop temperature change. Given our finding that people do not typically vote 0 when comfortable, we chose the minimum change as our strategy and save exploring other strategies for future work. We show that by not choosing input scales that reduce bias, we improve the response time of system by 86%.

Table 9.8 shows the time to converge to an acceptable comfort range (-0.5 to 0.5 per ASHRAE standards) and the standard deviation from neutral 0 for rooms with multiple occupants. Figure 9.15 shows the average PMV over time. The three point scale takes 17.7 days to reach the comfortable range. A longer time is required to reach the comfort range since the temperature corrections are smaller for each voting period. As expected, the 5 point range takes 11.6 days to reach the comfort range. The 7 point scale only took 9.5 days to converge. To validate the method, we compare the simulated 7 point results with a similar area in our live deployment. We see that both simulated and actual results are very similar showing 9.5 days and 10.0 days respectively. All three methods had very similar standard deviations. The similar deviation indicates the the bias actually does not seem to hinder the stability of the PMV. This is likely due to the fact that bias is "self-correcting" since occupants tend to vote at the extremes of the scale to maximize their influence. However, since the higher point scales allow for larger temperature corrections, the convergence for these strategies occurs faster.

Table 9.9 shows the time to converge for a single office using different scales. Again we find similar results to the multiple occupancy area. However, much less time is required to reach the target range since there is no competition with other occupants. In this case, the 7 point scale is preferable since there is no competition and the time required to reach the optimal comfort range is almost half (3.3 hours vs 6 hours). We again validate the simulation by comparing the simulated office with data from an actual office (3.3 hours vs 3.7 hours).

9.6 Discussion

In this section we discuss some of our experiences with our crowd sourcing application.

9.6.1 User Participation

One of the most critical components to a success to crowd sourcing is actual user participation. The quality, quantity, and reliability of the data is entirely in the hands of the users. Our main concerns were whether participants would be willing to provide baseline data for an entire week. Also a concern was whether the quantity of data would be enough to estimate AMV effectively. Overall, we found that participants were more than willing to provide votes; some participant actually voted every 15 min for the baseline week. Since occupants knew their votes during the learning period would affect their "thermal well-being", they were much more inclined to provide votes. Having a direct benefit to the user greatly

Point Scale	Convergence Time	Std. Dev.
3 points	6 hours	0.3533
5 points	4.5 hours	0.3402
7 points (Simulated)	3.3 hours	0.3171
7 points (Actual Avg.)	3.7 hours	0.2742

Table 9.9: Point scale results, single occupant.

increases success of the application, and important factor of a participatory sensing application.

9.6.2 Voting Bias

The vote variation shows that users can provide false information in order to better benefit. Users can give inflated estimates of their vote to give more weight to their preference in the hopes of increasing their thermal comfort. The bias present in our system did not affect overall satisfaction in our application, but this issue must be considered with other applications using crowd sourcing. More generally, one must consider the user's interest in the data and how they might artificially create bias in order to benefit. In particular, one must examine how this bias affects that target application.

9.6.3 User Psychology

In [Pac90], they showed that users with the illusion of some control more likely to be satisfied. We found that building managers do indeed take this to heart. We found most of thermostats on walls completely disabled from making temperature changes. However, we also found there is a limit to this rouse. 62.5% of our participants had access to a thermostat. However, 90% of these participants stated that the thermostat was ineffective. On other hand, if users actually do have some level of control, our results seem to confirm that this control greatly increases satisfaction and perceived comfort.

9.6.4 System Issues

In certain situations it is not efficient to cycle a thermostat constantly. However, this typically refers to large changes over relatively short periods for standard package units or any other system that is forced to react relatively fast to meet temperature demands. Our strategy only effects small changes every 10 minutes. In many cases, the period between changes is even larger. In addition, since the building uses district cooling using thermal storage rather than a package unit, our system is not affect by frequent temperature cycling; frequent temperature changes do not cycle the chillers that supply the cold water to the thermal storage tank. Another potential system issue is if a window is open. For most areas, this will automatically turn off the HVAC system to the area. This is part of the sequence of operations done in order to save energy. For other areas without window switches, this will mean that the room will lose additional conditioned area, decreasing efficiency. From a comfort perspective, this is simply the user "correcting" their PMV estimate using a parameter not available by our crowd sourced system.

9.6.5 Limitations

Crowd sourcing of thermal comfort can be affected by creating an appropriate incentive structure. For example, one could motivate users by positive (e.g. green points, future QOS in heat/cold waves, cash, etc.) and/or negative (e.g. extra fees, social pressure by public outing, etc.) incentives. For example, users may choose to wear a sweater if they are cold, instead of voting cold in order to save energy. While this behavior would affect the voting patterns, we believe incentive structures could be readily integrated into our crowd sourcing platform by providing user incentive tracking/accounting and even modeling of users' responses to incentives. We have left the exploration of this issue for future work.

9.7 Summary

In this chapter, we developed a crowd sourcing platform called Thermovote. This system uses cell phone applications and a website to allow users to provide feedback regarding their thermal comfort. Utilizing this platform we created two applications. The first are two live long term (25 and 15 months) deployments of a thermal comfort driven temperature control. We develop two different methods of utilizing Thermovote data for controlling the temperatures of rooms and a method of correcting PMV estimates in order to determine temperature changes. Our long term deployments showed significant improvement over baseline. With a 16% increase of energy, we were able to achieve 80% satisfaction across all testing areas. Our second application was the development of non-parametric thermal comfort models that can be integrated into an energy simulator. Through simulations we showed that optimizing comfort increases energy from 4%-16% depending on the season. We also explored potential energy savings by relaxing thermal preference requirements. By maintaining an AMV of ±0.7, we show annual savings of 13%. We also show through validated simulations how other voting scales would affect temperature convergence. With these applications, we have demonstrated a novel tool for examining the interaction of thermal comfort with building conditions constructed with Thermovote data. This work is an initial foray into the design of crowd sourcing mechanisms for control of complex engineering systems. Our models, simulations and experiments represent a non-trivial exploration of the problem space. Such techniques will find increasing importance as the community seeks ways to exploit the information offered by humans as a way of addressing new dimensions of building performance such as comfort levels and energy savings.

CHAPTER 10

Conclusion and Future Work

There is tremendous opportunity for reducing energy in buildings by conditioning to the appropriate level. We explored four areas central to usage based occupancy control that utilize wireless sensor technology; occupancy estimation, occupancy prediction, actuation using real-time and predicted occupancy, and user driven temperature control. Measuring occupancy is a critical for determining adequate conditioning for a room. With real-time occupancy information, it is possible to develop conditioning strategies for reducing energy costs by adjusting the ventilation and relaxing the set-points of a room based on occupancy. However, knowledge of occupancy alone is not sufficient conditioning for temperature. Since, time is required for the room temperatures to reach target levels, occupancy prediction is necessary in order to determine when to begin conditioning a room. While reducing energy is an important goal, just as critical is the ability of the control strategies to maintain comfortable conditions for the spaces. This dissertation examines each of these issues and how how they can be integrated together.

We started by examining occupancy prediction. Since rooms cannot be actuated instantaneously and take time to reach target temperatures, we present several occupancy prediction methods based on inhomogeneous Markov Chains in order to determine how far in advance rooms should be conditioned. We show that the size of the state space can be reduced by using only states observed during training and that discontinuities in the chain can be prevented using by a closest distance Markov chain (CDMC) approach, or by continuously blending the transition matrices over time (BMC). We demonstrate that the models capture the occupancy dynamics accurately when comparing samples generated from the model with ground-truth data using various metrics. With these prediction models, we then developed conditioning strategies for conditioning spaces based on predicted occupancy patterns. We show through both simulations and real-world deployments that conditioning temperature and ventilation based on actually occupancy can save up 42% or more annually depending on the climate.

In this dissertation, two different hardware platforms were developed for measuring occupancy; a low-power wireless camera platform deployed in public hallways and a wireless sensor thermal sensing platform that can be deployed in rooms. We showed that the camera based system is capable of detecting transitions with up to 94% accuracy and can be generalized to different locations. Using a particle filter, we bounded the error of occupancy within 1.83 people on average for our building. When inexpensive thermal sensing became available we developed a thermal sensing platform to be even more accurate and less power intensive. With this system we are able to detect occupancy with a RMSE of ≈0.35 persons. We showed it is possible to use low resolution thermal sensing to determine how many people are within the space by utilizing a novel occupancy classification process. Unlike the camera system, the thermal array is not affected by cumulative error. We increased accuracy of detecting empty spaces and the overall accuracy of the platform by also including a PIR sensor. This PIR sensor also helps to reduce the node's power consumption by triggering sensing only when someone is present. We showed both of these strategies to be successful for measuring occupancy within rooms in near real-time.

In order to address the issue of comfort, we developed a crowd sourcing platform called Thermovote. Users are able to use a cell phone application or website to provide feedback regarding their thermal comfort. We develop two different methods of utilizing Thermovote data for temperature actuation and a method of correcting PMV estimates in order to determine temperature changes. We also developed novel tool for examining the interaction of thermal comfort with building conditions constructed with system data. We use non-parametric thermal comfort models that can be integrated into an energy simulator and explored potential energy savings by relaxing thermal preference requirements.

There are multiple areas where additional future work is possible. Learning the thermal properties of the room is one such optimization. By taking into account the parameters such as current air temperature, damper position, and discharge air temperature, it would be possible to determine a more optimal ramp up time. The prediction models for PMV developed in the last chapter could be integrated into control strategies. Perhaps the most ambitious project would be integrating occupancy, thermal preference, and the thermal properties of the building, and HVAC components into a optimization problem that minimizes energy consumption while maximizing occupant comfort.

References

[ABD11] Y. Agarwal, B. Balaji, S. Dutta, R. K. Gupta, and T. Weng. "Duty-cycling buildings aggressively: The next frontier in HVAC control." In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pp. 246–257, April 2011.

[AC01] Bass Abushakra and David Claridge. "Accounting for the Occupancy Variable in Inverse Building Energy Baselining Models." In *ICEBO: Proceedings of the First International Conference for Enhanced Building Operations*, Austin, TX, USA, 2001.

[ACF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. "Finite-time Analysis of the Multiarmed Bandit Problem." *Mach. Learn.*, **47**(2-3):235–256, May 2002.

[AD95] Centre for the Analysis and Dissemination of Demonstrated Energy Technolgies. "CADDET, Saving energy with efficient lighting in commercial buildings." *Maxi Brochure 01*, 1995.

[ASH07a] "ASHRAE Standard 55: Thermal Environmental Conditions for Human Occupancy." American Society of Heating, Refrigeration and Air-Conditioning Engineers,Inc., 2007.

[ASH07b] "ASHRAE Standard 62.1: Ventilation for Acceptable Indoor Air Quality." American Society of Heating, Refrigeration and Air-Conditioning Engineers,Inc., 2007.

[ASH07c] "ASHRAE Standard 90.1: Energy Standard for Buildings Except Low-Rise Residential Buildings." American Society of Heating, Refrigeration and Air-Conditioning Engineers,Inc., 2007.

[ASP97] A. Auliciems, S. V. Szokolay, Passive, Low Energy Architecture International, International PLEA Organisation, and University of Queensland. Dept. of Architecture. *Thermal Comfort*. Design tools and techniques. PLEA, 1997.

[AZHa] Edward Arens, Hui Zhang, and Charlie Huizenga. "Partial and whole-body thermal sensation and comfort, Part I: Uniform environmental conditions." In *UC Berkeley: Center for the Built Environment*.

[AZHb] Edward Arens, Hui Zhang, and Charlie Huizenga. "Partial and whole-body thermal sensation and comfort, Part II: Uniform environmental conditions." In *UC Berkeley: Center for the Built Environment*.

[bed11] *2010 Building Energy Data Book*. U.S. Dept. of Energy, 2011.

[BET08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features." In *CVIU*, 2008.

[BFB93] G. Brager, M. Fountain, C. Benton, Edward A. Arens, and Fred Bauman. "A Comparison of Methods for Assessing Thermal Sensation and Acceptability in the Field." *Energy and Buildings*, 1993.

[BJ90] George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, Incorporated, 1990.

[BK09] Bill Burke and Marian Keeler. *Fundamentals of integrated design for sustainable building.* Wiley, 2009.

[BLE11] Y. Benezeth, H. Laurent, B. Emile, and C. Rosenberger. "Towards a sensor for detecting human presence and characterizing activity." *Energy and Buildings*, 2011.

[Bui] eQuest Building Energy Analysis Tool. "http://www.doe2.com/.".

[BXN13] Bharathan Balaji, Jian Xu, Anthony Nwokafor, Rajesh Gupta, and Yuvraj Agarwal. "Sentinel: Occupancy Based HVAC Actuation Using Existing WiFi Infrastructure Within Commercial Buildings." In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pp. 17:1–17:14, New York, NY, USA, 2013. ACM.

[CAB08] P. Chen, P. Ahammad, C. Boyer, Shih-I Huang, Leon Lin, E. Lobaton, M. Meingast, Songhwai Oh, S. Wang, Posu Yan, A. Y. Yang, Chuohao Yeo, Lung-Chung Chang, J. D. Tygar, and S. S. Sastry. "CITRIC: A low-bandwidth wireless camera network platform." In *Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2008)*, pp. 1–10, September 2008.

[CB04] E. F. Camacho and C. Bordons. *Model Predictive Control.* Advanced Textbooks in Control and Signal Processing. Springer, 2004.

[CBS] Dong Chen, Sean Barker, Adarsh Subbaswamy, David Irwin, and Prashant Shenoy. "Non-Intrusive Occupancy Monitoring Using Smart Meters." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys 2013.

[Com93] California Energy Commission. "Advanced Lighting Guidelines." *CEC Publications*, 1993.

[DA09] Bing Dong and Burton Andrews. "Sensor-Based Occupancy Behavioral Pattern Recognition for Energy and Comfort Management in Intelligent Buildings." In *International Building Performace Simulation Association*, 2009.

[DC02] D. T. Shadish, W.R., Cook, T.D. and Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference.* Houghton-Mifflin, 2002.

[DHT06] Robert H. Dodier, Gregor P. Henze, Dale K. Tiller, and Xin Guo. "Building occupancy detection through sensor belief networks." *Energy and Buildings*, **38**(9):1033–1043, 2006.

[Doe12] "DOE-2 - Building Energy Analysis Tool and Cost Analysis Tool." `http://www.doe2.com/DOE2`, 2012.

[EBV13] Afrooz Ebadat, Giulio Bottegal, Damiano Varagnolo, Bo Wahlberg, and Karl H. Johansson. "Estimation of Building Occupancy Levels Through Environmental Signals Deconvolution." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pp. 8:1–8:8, New York, NY, USA, 2013. ACM.

[EC] Varick Erickson and Alberto E. Cerpa. "Thermovote: Participatory Sensing for Efficient Building HVAC Conditioning." In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2012).*

[ECC11] V. L. Erickson, M. A. Carreira-Perpinan, and A. E. Cerpa. "OBSERVE: Occupancy-based system for efficient reduction of HVAC energy." In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pp. 258–269, April 2011.

[Epl] "EnergyPlus - Building Energy Analysis Tool." http://apps1.eere.energy.gov/buildings/energyplus/.

[EPS01] Steven J. Emmerich, Andrew K. Persily, National Institute of Standards, Technology (U. S.), and Architectural Energy Corporation. *State-of-the-art review of CO2 demand controlled ventilation technology and application.* U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, [Gaithersburg, Md.] :, 2001.

[eur] "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data." The European Parliament and the Council of the European Union.

[FFS06] William J. Fisk, David Faulkner, and Douglas P. Sullivan. "Accuracy of CO2 sensors in commercial buildings: a pilot study." Technical Report Technical Report LBNL-61862, Lawrence Berkeley National Laboratory, 2006.

[FP02] David A. Forsyth and Jean Ponce. "Computer Vision: A Modern Approach." 2002.

[Gag86] L. G. Gagge, A.P., Fobelets, A.P., Berglund. "A Standard Predictive Index of Human Response to the Thermal Environment." *ASHRAE Transactions*, 1986.

[GB00] Vishal Garg and N. K. Bansal. "Smart occupancy sensors to reduce energy consumption." *Energy and Buildings*, pp. 81–87, 2000.

[GK] Peter Xiang Gao and S. Keshav. "Optimal Personal Comfort Management Using SPOT+." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys 2013.

[GKJ12] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. "Combinatorial Network Optimization with Unknown Variables: Multi-armed Bandits with Linear Rewards and Individual Observations." *IEEE/ACM Trans. Netw.*, **20**(5), October 2012.

[gri] "Grid-Eye - Infared Array Sensor.".

[Ham94] J. D. Hamilton. *Time Series Analysis.* Princeton University Press, 1994.

[Hed04] Alan Hedge. "Linking environmental conditions to productivity." *Eastern Ergonomics Conference and Exposition*, 2004.

[HL83] D. Herron and Construction Engineering Research Laboratory. *Use of simplified input for BLAST energy analysis.* Construction Engineering Research Laboratory, U.S. Army, Corps of Engineers; National Technical Information Service, 1983.

[HLJ09] Nathan R. Hoot, Larry J. LeBlanc, Ian Jones, Scott R. Levin, Chuan Zhou, Cynthia S. Gadd, and Dominik Aronsky. "Forecasting Emergency Department Crowding: A Prospective, Real-time Evaluation." *Journal of the American Medical Informatics Association*, 2009.

[Hop99] P. Hoppe. "The physiological equivalent temperature - a universal index for the biometeorological assessment of the thermal environment." *International Journal of Biometeorology*, 1999.

[HT73] John Hopcroft and Robert Tarjan. "Algorithm 447: efficient algorithms for graph manipulation." *Commun. ACM*, June 1973.

[HW] Lam Abraham Hang-yat and Dan Wang. "Carrying My Environment with Me: A Participatory-sensing Approach to Enhance Thermal Comfort." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys 2013.

[HYC04] Edwin O. Heierman, G. Michael Youngblood, and Diane J. Cook. "Mining temporal sequences to discover interesting patterns." In *In: Proceedings of the 2004 International Conference on Knowledge Discovery and Data Mining*, 2004.

[HZA] Charlie Huizenga, Hui Zhang, Edward Arens, and Danni Wang. "Skin and core temperature response to partial and whole-body heating and cooling." In *UC Berkeley: Center for the Built Environment*.

[imb] "IntelMote2 and IMB 400 sensor board." http://www.memsic.com/.

[ISO06] *Ergonomics of the thermal environment. Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria.* ISO, 2006.

[Jaz11] Becerik-Gerber B. Jazizadeh F, Kavulya G, Klein L. "Continous Sensing of Occupant Perception of Indoor Ambient Factors." *ASCE Workshop of Computing in Civil Engineering*, 2011.

[KB] Mohamed-Bécha Kaâniche and Franccois Bremond. "Gesture Recognition by Learning Local Motion Signatures." In *CVPR 2010 : IEEE Conference on Computer Vision and Pattern Recognition.*

[KB11] Anthony Kelman and Francesco Borrelli. "Bilinear model predictive control of a HVAC system using sequential quadratic programming." In *IFAC World Congress*, 2011.

[KBS13] Wilhelm Kleiminger, Christian Beckel, Thorsten Staake, and Silvia Santini. "Occupancy Detection from Electricity Consumption Data." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, Roma, Italy, 2013. ACM.

[kin] "Microsoft Kinect." www.microsoft.com/en-us/kinectforwindows/.

[KJD09] Ankur Kamthe, Lun Jiang, Matt Dudys, and Alberto Cerpa. "SCOPES: Smart Cameras Object Position Estimation System." In *In the Proceedings of the 6th European Conference on Wireless Sensor Networks(EWSN 2009)*, pp. 279–295, Cork, Ireland, February 2009. Springer-Verlag.

[LBS09] D. Lymberopoulos, A. Bamis, and A. Savvides. "The BehaviorScope Framework for Enabling Ambient Assisted Living." In *Personal and Ubiquitous Computing*, 2009.

[LHD09] Khee Poh Lam, Michael Hoynck, Bing Dong, Burton Andrews, Yun-Shang Chiou, Rui Chang, Diego Benitez, and Joonho Choi. "Occupancy Detection Through an Extensive Environmental Sensor Network in an Open-Plan Office Building." In *International Building Performace Simulation Association*, 2009.

[LM02] R. Lienhart and J. Maydt. "An extended set of Haar-like features for rapid object detection." In *ICIP*, 2002.

[Log] Automated Logic. "`http://www.automatedlogic.com/`.".

[Low99] D. Lowe. "Object recognition from local scale-invariant features." In *ICCV*, 1999.

[LSS10] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. "The smart thermostat: using occupancy sensors to save energy in homes." In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pp. 211–224. ACM, 2010.

[MAA01] Gopal P Maheshwari, Hanay Al-Taqi, Raba'a Al-Murad, and Rajinder K Suri. "Programmable thermostat for energy saving." *Energy and Buildings*, pp. 667–672, 2001.

[Mit97] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[MJK] Parisa Mansourifard, Farrokh Jazizadeh, Bhaskar Krishnamachari, and Burcin Becerik-Gerber. "Online Learning for Personalized Room-Level Thermal Control: A Multi-Armed Bandit Framework." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys 2013.

[MKS04] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. "The flooding time synchronization protocol." In *SenSys*, 2004.

[MMC11] S. Mamidi, R. Maheswaran, and Y. Chang. "Smart Sensing, Estimation, and Prediction for Efficient Building Energy Management." In *Multi-agent Smart Computing Workshop*, 2011.

[MR81] C. S. Myers and L. R. Rabiner. "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition." *Bell System Technical Journal*, **60**(7), 1981.

[MRN11] R. Melfi, B. Rosenblum, B. Nordman, and K. Christensen. "Measuring building occupancy using existing network infrastructure." *International Green Computing Conference and Workshops*, 2011.

[Ngh] Truong X. Nghiem. "MLE+: a Matlab-EnergyPlus Co-simulation Interface." http://www.seas.upenn.edu/ nghiem/mleplus.html.

[NLP98] "Occupancy Sensors, Motion-sensing devices for lighting control." *Lighting Research Center*, 1998.

[O70] Fanger P. O. *Thermal comfort. Analysis and applications in environmental engineering.* Danish Technical Press, 1970.

[OEC] "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data." Organization for Economic Cooperation and Development.

[OP02] Olesen, B. W. and Parsons, K. C. "Introduction to thermal comfort standards and to the proposed new version of EN ISO 7730." *Energy & Buildings*, July 2002.

[Ope] OpenCV. "http://opencvlibrary.sourceforge.net/.".

[Pac90] Monica Paciuk. *The Role of Personal Control of the Environment in Thermal Comfort and Satisfaction at the Workplace.* Environmental Design Research Association, 1990.

[PaP] "Panasonic PIR." http://pewa.panasonic.com/downloads/papirs-ekmb/.

[PEI] "Personal Environmental Impact Report.".

[PHK07] Jens U. Pfafferott, Sebastian Herkel, Doreen E. Kalz, and Andreas Zeuschner. "Comparison of low-energy office buildings in summer using different thermal comfort criteria." *Energy and Buildings*, pp. 750–757, 2007.

[PHK10] Joern Ploennigs, Burkhard Hensel, and Klaus Kabitzsch. "Wireless, collaborative virtual sensors for thermal comfort." In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys, pp. 79–84. ACM, 2010.

[pho] "Phoenix Controls." `www.phoenixcontrols.com/`.

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, USA, 1994.

[RA09] M. Rawi and A. Al-Anbuky. "Passive House sensor networks: Human centric thermal comfort concept." *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on*, 2009.

[RSD10] Sasank Reddy, Katie Shilton, Gleb Denisov, Christian Cenizal, Deborah Estrin, and Mani Srivastava. "Biketastic: Sensing and Mapping for Better Biking." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10. ACM, 2010.

[RTI08] Ian Richardson, Murray Thomson, and David Infield. "A high-resolution domestic building occupancy model for energy demand simulations." *Energy and Buildings*, 2008.

[SBK11] James Scott, A. J. Bernheim Brush, John Krumm, Brian Meyers, Mike Hazas, Stephen Hodges, and Nicolas Villar. "PreHeat: controlling home heating using occupancy prediction." In *UbiComp'11*, pp. 281–290, 2011.

[SBW10] M. Schumann, A. Burillo, and N. Wilson. "Predicting the desired thermal comfort conditions for shared offices." In *International Conference on Computing in Civil and Building Engineering (ICCCBE-10)*, 2010.

[SGS] Mani Srivastava, Jeffrey Goldman, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, Nithya Ramanathan, Sasank Reddy, Vids Samanta, and Ruth West. "Participatory Sensing - A citizen-powered approach to illuminating the patterns that shape our world.".

[Sol] WSK-24 Wireless Occupancy Solution. "`http://specifyhoneywell.com/`.".

[T10] "TEP105: Low Power Listening." http://www.tinyos.net/tinyos-2.x/doc/html/tep105.html.

[TBF06] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2006.

[TC07] Wai Leung Tse and Wai Lok Chan. "Real-time measurement of thermal comfort by using an open networking technology." *Measurement*, pp. 654–664, 2007.

[Tit13] "2013 Building Energy Efficiency Standards." `http://www.energy.ca.gov/title24/2013standards/`, 2013.

[tmo] "Tmote Sky." http://www.snm.ethz.ch/Projects/TmoteSky.

[tru] "TrueView People Counter." `http://www.cognimatics.com/`.

[TS07] Thiago Teixeira and Andreas Savvides. "Lightweight People Counting and Localization in Indoor Spaces Using Camera Sensor Nodes." *ICDSC*, 2007.

[TYS13] Kevin Ting, Richard Yu, and Mani Srivastava. "Occupancy Inferencing from Non-intrusive Data Sources." In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13. ACM, 2013.

[VJ01] P. Viola and M. Jones. "Robust Real-time Object Detection." In *International Journal of Computer Vision*, 2001.

[WFR05] Danni Wang, Clifford C. Federspiel, and Francis Rubinstein. "Modeling occupancy in single person offices." *Energy and Buildings*, pp. 121–126, 2005.

[WH03] Billy M. Williams and Lester A. Hoel. "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results." *Journal of Transportation Engineering*, 2003.

[YYC03] Guodong Ye, Changzhi Yang, Youming Chen, and Yuguo Li. "A new approach for measuring predicted mean vote (PMV) and standard effective temperature (SET)." *Building and Environment*, 2003.

[ZHA] H Zhang, C Huizenga, E Arens, and T Yu. "Modeling thermal comfort in stratified environments." In *UC Berkeley: Center for the Built Environment.*