POEM: Power-efficient Occupancy-based Energy Management System

Varick L. Erickson
Elect. Eng. & Comp. Science
Univ. of California, Merced
verickson@ucmerced.edu

Stefan Achleitner
Elect. Eng. & Comp. Science
Univ. of California, Merced
sachleitner@ucmerced.edu

Alberto E. Cerpa
Elect. Eng. & Comp. Science
Univ. of California, Merced
acerpa@ucmerced.edu

ABSTRACT

Buildings account for 40% of US primary energy consumption and 72% of electricity. Of this total, 50% of the energy consumed in buildings is used for Heating Ventilation and Air-Conditioning (HVAC) systems. Current HVAC systems only condition based on static schedules; rooms are conditioned regardless of occupancy. By conditioning rooms only when necessary, greater efficiency can be achieved. This paper describes POEM, a complete closed-loop system for optimally controlling HVAC systems in buildings based on actual occupancy levels. POEM is comprised of multiple parts. A wireless network of cameras called OPTNet is developed that functions as an optical turnstile to measure area/zone occupancies. Another wireless sensor network of passive infrared (PIR) sensors called BONet functions alongside OPTNet. This sensed occupancy data from both systems are then fused with an occupancy prediction model using a particle filter in order to determine the most accurate current occupancy in each zone in the building. Finally, the information from occupancy prediction models and current occupancy is combined in order to find the optimal conditioning strategy required to reach target temperatures and minimize ventilation requirements. Based on live tests of the system, we estimate $\approx 30.0\%$ energy saving can be achieved while still maintaining thermal comfort.

Categories and Subject Descriptors

J.7 [**Computers In Other Systems**]: Command & control; C.3 [**Special-Purpose and Application-Based Application Systems**]: Real-time and embedded systems

General Terms

Experimentation, Measurement, Performance

Keywords

Occupancy, HVAC, Ventilation, Energy savings

1. INTRODUCTION

Sustainable buildings represent a major paradigm shift from current practice. According to the US Department of Energy, buildings account for nearly 40% of US primary energy consumption in 2010, 72% of which is electrical energy [1]. Of this total, 50% of the energy consumed in buildings is used for heating, air-conditioning and ventilation (HVAC) systems [1]. Reducing this load is a priority if we wish to achieve energy independence. Within office buildings, one obvious source of inefficiency are the empty or partially filled rooms. Rather than assuming all rooms are used equally, rooms should be conditioned based on actual usage.

Several works have shown that approximately 25%-40% HVAC energy saving can be achieved by regulating HVAC systems based on occupancy [15, 11]. These papers discuss sensing solutions for detecting room usage. Passive infrared (PIR) sensors are commonly used for binary sensing of occupancy for lighting. These sensors, however, do not enable the adjustment of ventilation, which depends on the number of occupants. CO_2 sensors are another alternative. They can infer level of occupancy and ventilation rates but are slow to respond and suffer from calibration problems [16]. For our system, we utilize cameras deployed in public hallways along with PIR sensors within rooms to infer occupancy. Instead of deploying cameras directly in rooms to count individuals, we deploy cameras that count the number of people that pass across an area. By using cameras as optical turnstiles, it is possible to measure occupancy levels of areas connected to the turnstile area. 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 video data to a central location. For our experiments we follow well-known best practices with respect to the use of privacy-critical data [5, 2]. Furthermore, our distributed data processing solution enhances scalability and significantly reduces network load, improving system lifetime of the bandwidth-limited camera sensor network. Existing optical counting systems exist, but they are difficult to retrofit to buildings and require hard wiring [8]. Wiring can be very difficult for many older buildings where drop ceilings are not available. We wish to utilize motes that can be easily deployed in even older buildings. Maintenance time can be minimized by coordinating battery replacement with routine light-bulb replacement.

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's comfort, we must be able to predict when occupants are likely to enter a room and begin conditioning beforehand. We achieve this by using a blended Markov Chain as described in [15].

This paper contributes 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.

• Our most significant contribution is the design, implementation, deployment and evaluation of the POEM system, which is a full closed-loop system that conditions rooms based on occupancy on a *real office building*. To our knowledge, our system is the first to control both temperature and ventilation based on near real-time occupancy.

• 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.

After examining related work in Section 2, we begin in Section 3 with an overview of POEM, a demand-driven system for optimally controlling HVAC systems. In the following sections, we describe the different components of POEM. Section 4 describes OPTNet, a 22 wireless camera deployment, which is followed by Section 5 that describes a 40 PIR wireless sensors. We then show in Section 6 how a particle filter can be used to correct errors in occupancy. Section 7 shows the programmatic interface with the building management system (BMS) and the control algorithm used to optimally condition rooms based on current and prediction occupancy. In Section 8, we evaluate the potential energy savings and the conditioning effectiveness with respect to temperature and ventilation. We then show in Section 9 that the cost of the system can be amortized in approximately 6-10 months if applied to the entire building and finally we discuss our conclusions and future work in Section 11.

2. RELATED WORK

As previously mentioned, our most significant contribution is the development of a full closed-loop occupancy based control system in a non-residential building. Since there are few such deployments, we examine work related to each of our subsystems.

In [17], the authors developed a 16 sensor node camera deployment within the hallways of an office building. While the system was shown to be accurate at counting individual components, independent tests over 24 hours showed that the system is only able to capture $\approx 80\%$ [15] of transitions. In particular, they do not address how errors in transitions lead to cumulative errors in occupancy. Our paper examines this problem in detail and how it can be addressed using a particle filter.

In [22], the authors monitor elderly people using Imote2 motes with Enalab cameras. To track occupants, they utilize a motion histogram. The system assumes that the camera can actively monitor the entire area of interest. The problem of privacy also exists for this system; cameras must be placed directly in the room.

The authors of [11] developed and tested an occupancy-based conditioning system. For their deployment they utilize PIR and door sensors to determine if rooms are occupied and condition rooms reactively. Ventilation is not considered as part of the conditioning strategy, which impacts energy efficiency. One of the main issues of their system evaluation is the conditioning effectiveness; they do not consider the comfort of their occupants during the ramp-up period for their reactive system. Our evaluation examines in detail the thermal comfort of occupants and also considers ventilation.

Door/PIR sensors are also used in [21] for conditioning residential buildings. They utilize a Hidden Markov Model in order to determine the probability that the house is occupied, occupied with occupants asleep, or unoccupied. Ventilation is not taken into account.

In [15], the authors develop an occupancy prediction model using a blended Markov Chain. In addition, they developed a conditioning strategy utilizing the occupancy prediction model, in simulation, to estimate energy savings. While these simulations provided analysis of ventilation, the paper does not implement the model in an actual deployment. The authors also describe agent based, multivariate Gaussian, and moving window Markov Chain occupancy models in [14, 13]. The main drawback of these works is that the performance of the proposed systems is done only in simulation. In particular, it is questionable whether the unbiased error introduced into the simulated sensor data is valid; in our deployments we found that errors in particular sensor locations can be bias. We show how to correct errors and verify energy saving with an actual deployment.

The authors of [25] developed a prediction based conditioning strategy for residential buildings using k-nearest-neighbors (KNN) to predict occupancy and a deployment utilizing PIR and RFID. This prediction method was 80% to 86% accurate for a 90 minute prediction, but did not not perform well for rooms that were not consistently occupied. While the approach is appropriate for residential buildings where the usage dynamic is simple, this approach may not work as effectively for office spaces where inter-room correlations exist. They also only consider temperature for their control, and do not consider ventilation.

3. POEM OVERVIEW

The main contribution of this work is POEM, which is a complete system that controls the temperature and ventilation of a building. This system is comprised of several components. Figure 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. 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
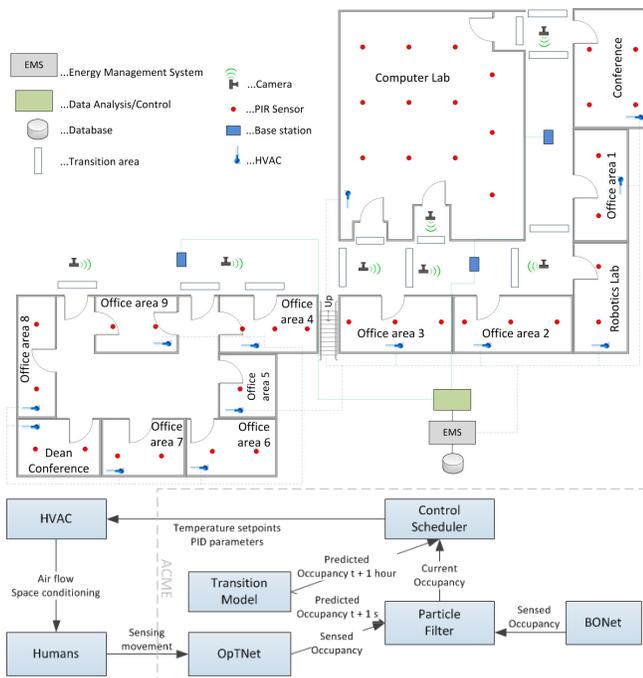
Figure 1: Architecture of the POEM system.

process is complete, the conditioning affects occupants completing the feedback loop. The next sections provide a detail description and evaluation of each POEM component.

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 4.1. In Section 4.3, we describe a lightweight image processing method that can be used with a classification algorithm to detect transitions.

4.1 Overview and Design Challenges

We use the Imote2 platform developed by Intel and the IMB400 Camera [4] 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 2, right), we are able to roughly view 9 m^2 when the cameras are deployed at a height of 3 m. 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 and thus save energy. Also important is the ventilation of the room, which is proportional to occupancy.

There are several design challenges. Motes have limited

Figure 2: Transition areas (bigger blue boxes) and trigger areas (small red boxes).

resources available. Though the IMB400 cameras are able to capture 32 fps, the Imote2 radio is only able to transmit ≈ 100 kbps under typical conditions, making near 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 [20] and SURF [12]. 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.

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

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.

Step 1: Transition Sequence Delimitation
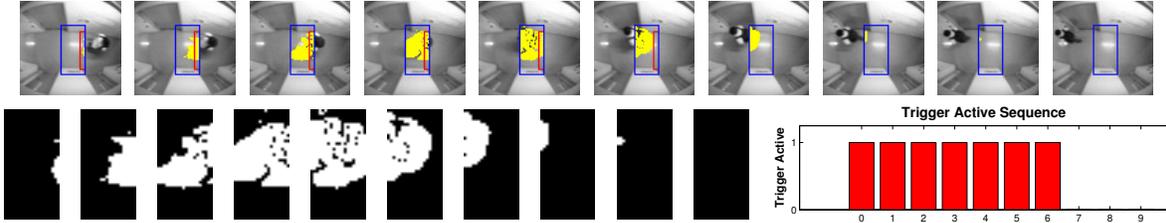
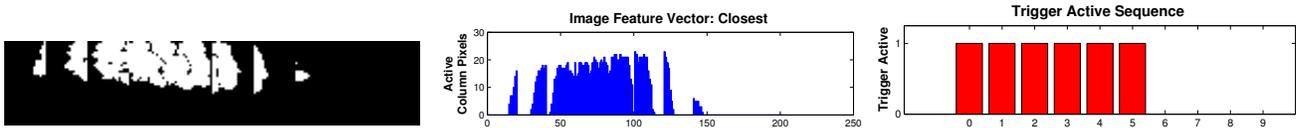**Step 2: Construction of Image and Trigger Sequence Feature Vectors**
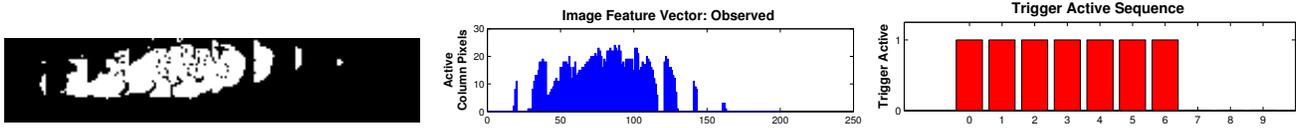
Step 3: KNN Match

Figure 3: Above is a summary of transition classification procedure.

Individually Trained Boundary Area

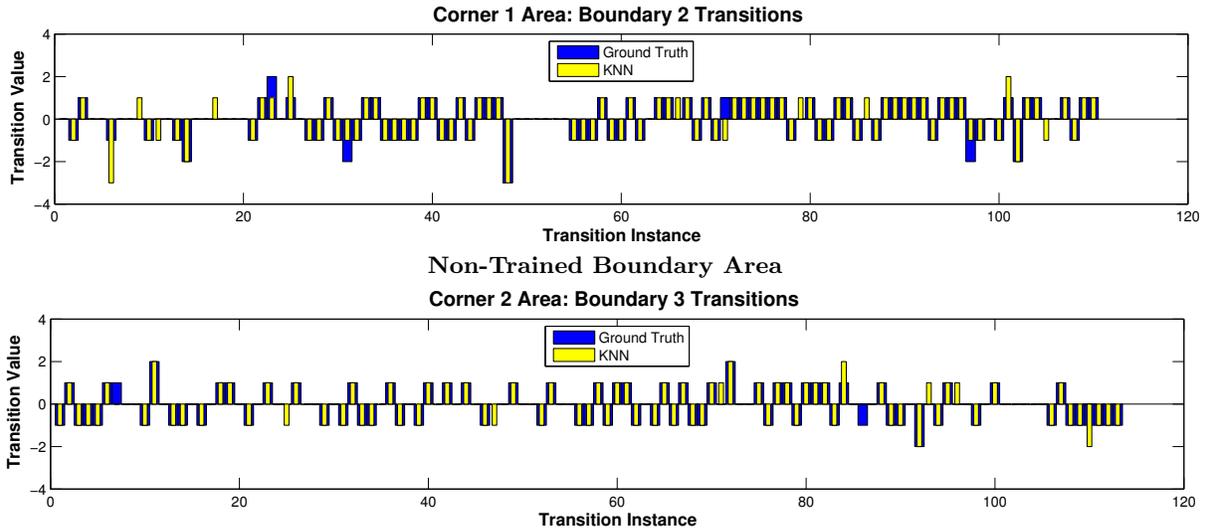**Non-Trained Boundary Area**

Figure 4: Ground Truth compared with KNN model for two different transition areas.
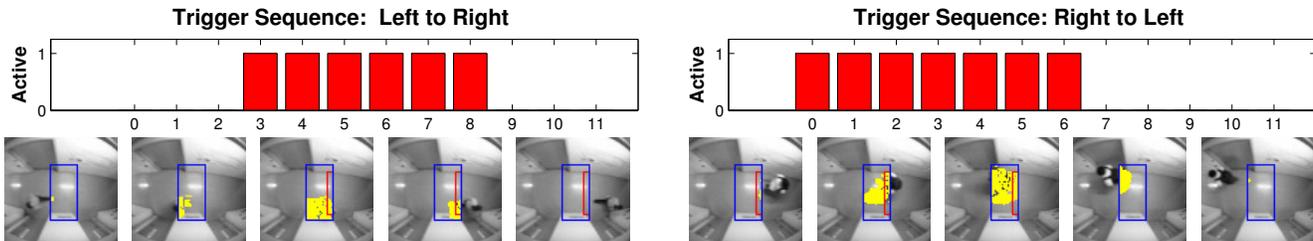

Figure 5: The sequence of the trigger state helps indicate direction. The red box indicates the trigger state is active.

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.

4.3.1 Motion Detection

For human motion detection we use previous frame background subtraction with a static threshold [**?**]. 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 pixels erroneously classified as active. 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.

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 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 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 5). Figure 3 shows a transition example. If a transition area has been empty for 3 previous frames (≈ 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 (≈ 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.

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 $[f_0 \ldots f_n]$ 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, $f_0 \ldots f_n$ 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 $v_i = \sum c_i$. 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 5 shows how the sequence of the trigger state can be used to infer direction. 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, more data would be sent. 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 be roughly put in 2 packets using the node RF transceiver mentioned in Section 4.1. Further compression could be achieved by aggregating groups of columns. This along with other compression techniques are left for future work.

4.3.4 KNN Transition Classification

K-nearest neighbors is an effective classification technique when few labeled samples are available. 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,

$$d(X, Y, Xt, Yt) = \sum_i |x_i - y_i| + \alpha \sum_i |xt_i - yt_i| \quad (1)$$

where α is a weighting coefficient. This coefficient helps to favor samples 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 $\pm 20\%$. If $m > n$ then the distance is defined as,

$$d(X, Y, Xt, Yt) = \operatorname*{argmin}_{off} \sum_{i=off}^{m-off} |x_i - y_i| + \alpha \sum_{i=off}^{m-off} |xt_i - yt_i| \quad (2)$$

where off is an offset that minimizes the distance when the smaller vector subtracted from a subset of the larger vector. 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 OPTNet Evaluation

Next we evaluate the accuracy of the algorithm. For each boundary, we gathered 48 hours of ground truth transition
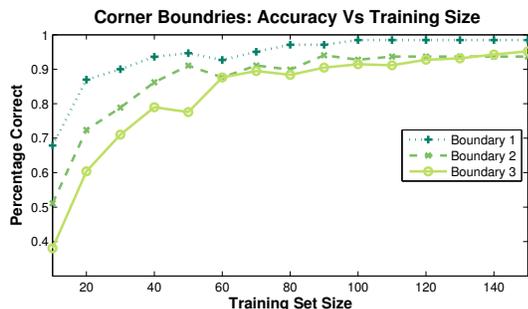
Figure 6: Overall accuracy over 24 hours as a function of training set size.

Figure 7: A single transition error continues to affect occupancy estimate.

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 $\alpha = 10$ 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 [24, 27, 18] 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 <u>and</u> number of occupants are classified correctly divided by the total number of transitions. Figure 4 shows the 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 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 6 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
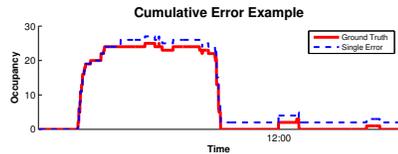
overall accuracy over 24 hours. Only 90 samples (roughly 45 each direction) are required to achieve above 90% directional accuracy. We found similar results for the other transition areas. Transition areas near corners or highly variable lighting tended to have lower accuracy. In these cases, additional training data would increase the accuracy as suggested by Figure 6. 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 receive more and better quality training examples for transitions through hallways than for transitions directly in front of doors.

4.4.1 Occupancy Error

In order to evaluate the error of the system, we collected 15 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 ($\approx 150,000$) 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 7 shows the effect a single transition error has on occupancy. Some basic strategies can be employed to reduce the magnitude of this cumulative error such as assuming negative occupancies to be 0, imposing limits to maximum occupancy, or assuming building occupancy to be 0 early in the morning (4am). While this helps to reduce occupancy error, if a room has a positive occupancy bias, it is possible to have long stretches of time where a room is marked as occupied when it is empty, which has a significant impact on energy saving strategies. To address this issue, we use a PIR wireless network deployment to help identify empty rooms along with a particle filter.

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 8). 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 [7] and timestamped using Flooding Time Synchronization Protocol [23]. 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
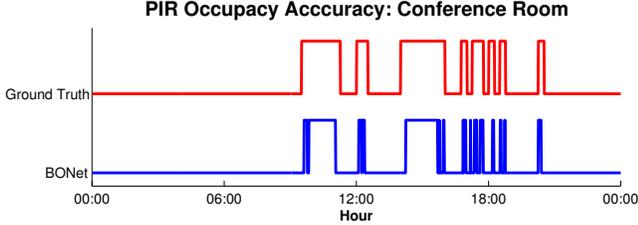
Figure 8: PIR node deployed on ceiling

Figure 9: PIR accuracy compared with ground truth.

sensors were available. Figure 9 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.

6. PARTICLE FILTER

We implemented a particle filter to improve the occupancy estimation. The particle filter algorithm is a nonparametric implementation of the Bayes filter [26]. The particle filter algorithm represents 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 assigns weights according to a measurement model representing the sensor accuracy. These random samples are called particles, which are used to estimate the distribution of the posterior state. Particles are defined as $X_t = x_t^1, x_t^2, \ldots x_t^M$ where, each particle x_t^m is a vector representing a set of N rooms. 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 1 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 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** draws M samples from the transition model. The transition model is represented by a blended Markov Chain as described in [15] (see Appendix), where each state x_{t-1}^m can reach one possible successor state $\hat{x}_t^m$ with a certain probability $p(\hat{x}_t^m | x_{t-1}^m), m = 1 \ldots M$.
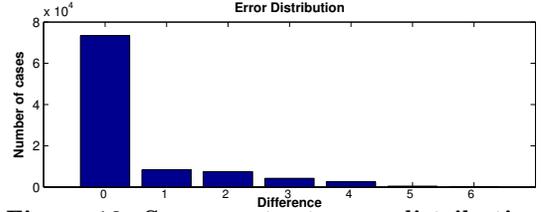

Figure 10: Sensor output error distribution.

Algorithm 1 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($w_t^{m,n}$), $m = 1 \ldots M, n = 1 \ldots N$
  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

The **weight calculation step** assigns a specific weight $w_t^{m,n}$ to each value of each state given the estimation probability $\hat{x}_t^{m,n}$ under the sensor measurement z_t^n, for a specific particle m of room n. In our case, the weight $w_t^{m,n}$ is determined by the measurement model, 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 (see Figure 10).

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; $x_{tDiff}^{m,n} = \hat{x}_t^{m,n} - z_t^n$. 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 difference $x_{tDiff}^{m,n}$ by the total 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 5240 out of 864k cases, which is a weight of 0.06. This is done for every room n within each state m of 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 particle 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.

$$CurrOcc = \frac{\sum_{m=1}^{M} x_t^{m,n}}{M}, \forall n \in N \quad (3)$$

6.1 Particle Filter Results

Figure 11 shows the occupancy estimation by a particle filter, the direct sensor output of the OPTNet system. The
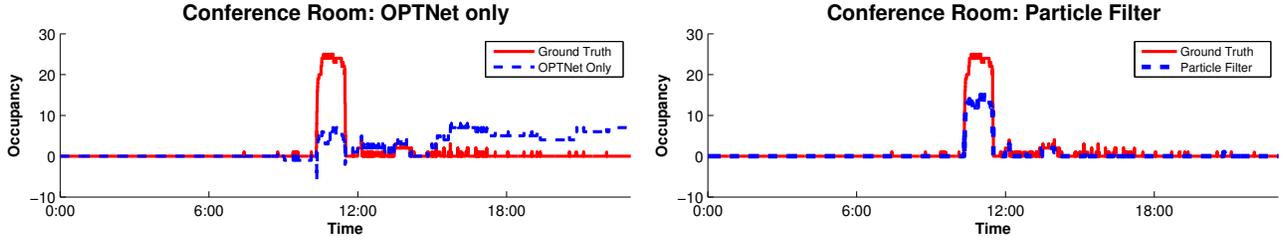
Figure 11: Occupancy over time for the raw sensor output and particle filter.
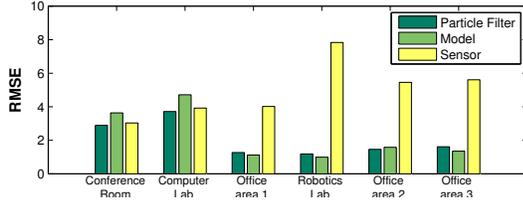

Figure 12: Occupancy error for 6 areas (2 weeks).

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 OBNet 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.

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 12 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.

7. ACTUATION SCHEDULER INTERFACE

The building being controlled utilizes both Automated Logic and Phoenix BMS systems [19, 6]. The system is maintained through a web server/interface called WebCtrl [19], which can issue BACnet commands 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.

7.1 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, determined the maximum amount of time it was required to reach set-point given the setbacks. Occupancy is predicted using the blended Markov Chain approach used in Section 6 (see Appendix). The main difference in application of the model is that predictions are done 1 hour in advance rather than the 1 second into the future done in for the particle filter. This is done since time is required in order for HVAC systems to react.

Algorithm 2 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

Before we discuss actuation, we must first discuss American Society of Heating, Refrigerating and Air-Conditioning (ASHRAE) standards for thermal comfort. ASHRAE Standard 55 [9] 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 temperatures where $-0.5 \leq PMV \leq 0.5$.

Algorithm 2 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 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.

8. 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 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 [10]. 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 [5, 2].

Table 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 [3] 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

	Area	Ramping 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 1: Zone information for deployment. The Ramping duration is the time required to reach the target temperature from the setback temperature.

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.

8.1 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 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 lost/gained by the air. We can estimate the energy consumed by a VAV or AHU using the heat balance equation $Q = mC(T_{sa} - T_{da})$, 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.

8.2 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
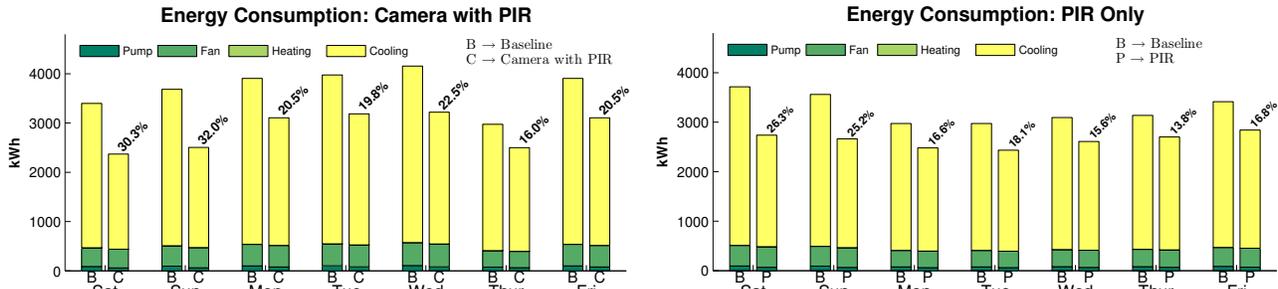
Figure 13: Energy consumption breakdown for 7 representative days. PIR and camera/PIR based POEM saves on average 21.1% to 26.0% respectively.
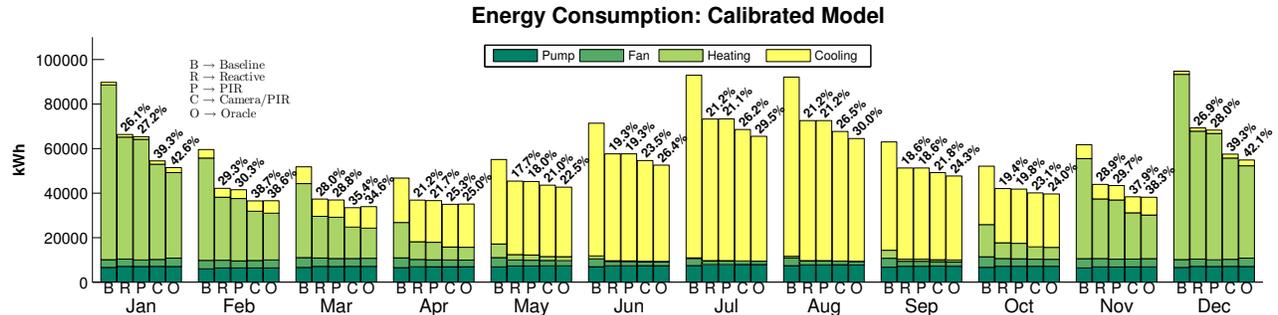

Figure 14: Energy consumption based on calibrated energy simulation. On average PIR and Camera based strategies save 24.5% and 31.0% annually respectively.

without POEM running. Days with similar weather and temperature profile are used for the baseline comparison.

Figure 13 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 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 condition. This is similar to trying to cool a house down on a hot summer day with the front door open.

8.3 Calibrated Model Results

As previously mentioned we also test four different strategies using the calibrated model. Figure 14 show the energy savings and breakdown for each strategy. When we compare the live 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
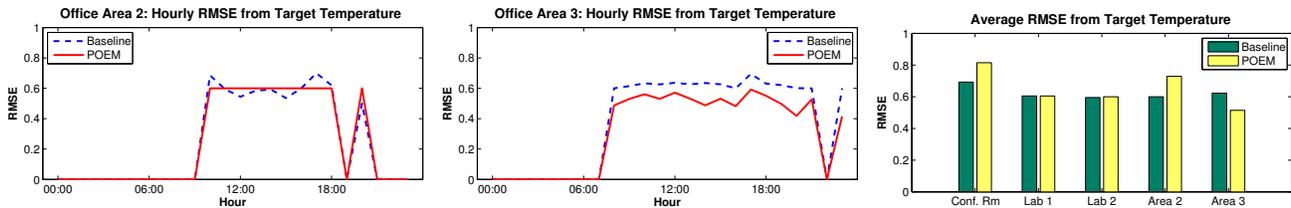
Figure 15: RMSE (F°) between the target and observed room temperatures

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%. 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 17 where we see that the PIR/Camera strategy is above the required ventilation rate for the majority of the time.

8.4 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 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 15 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
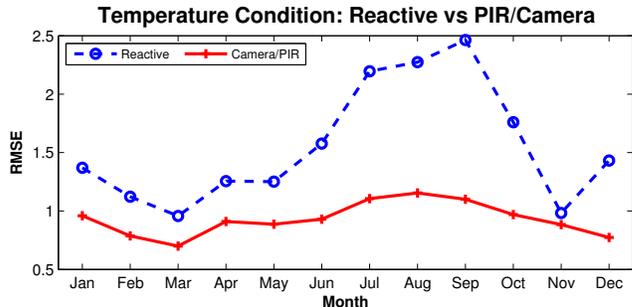

Figure 16: RMSE (F°) between the target and observed room temperatures from the calibrated model.

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 16 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 [10] requires the outdoor ventilation to be $V_{bz} = R_p P_z + R_a A_z$ 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 17, 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 ac-
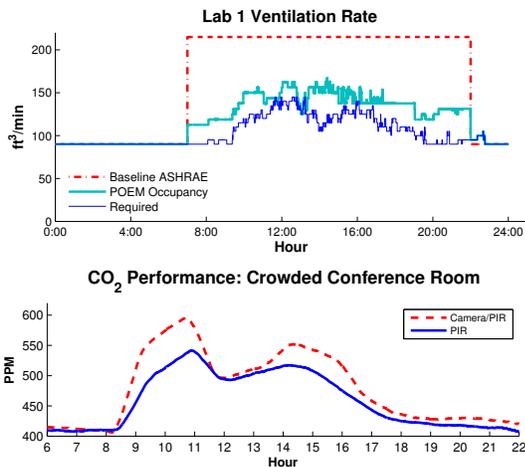
Figure 17: Ventilation and CO_2 levels

ceptable for office spaces. Figure 17 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.

9. 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 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

Table 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

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 18, 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 [1]), 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 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 18). Over the average lifetime of the building, which is approximately 70 years, the system can potentially save \$15.0M to \$20.0M.

10. 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 and legs [24, 27, 18] 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
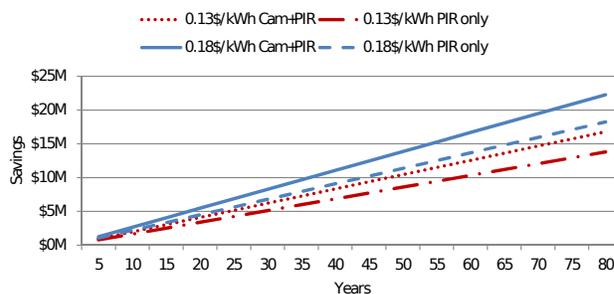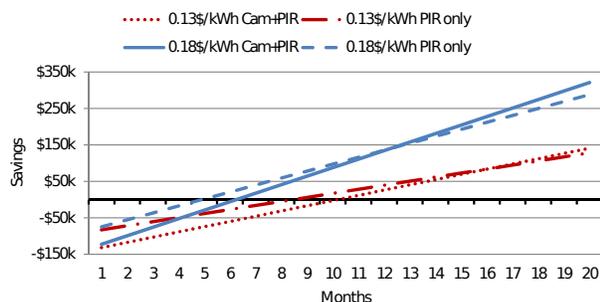
Figure 18: The left compares savings for different energy prices. The right shows savings over 80 year period.

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.

11. CONCLUSIONS

In this paper, our main contribution is the design and implementation of POEM, a close loop system that conditions rooms based on occupancy on a real production building. POEM uses two wireless sensor networks; one network of cameras called OPTNet, and another network of PIR sensors called BONet. We showed that OPTNet 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. 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 ≈ 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. This work is an initial foray into designing sensor networks for smart sustainable buildings. 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.

12. ACKNOWLEDGMENTS

13. REFERENCES

[1] *2010 Building Energy Data Book*. U.S. Dept. of Energy.
[2] *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.
[3] Energyplus - Building Energy Analysis Tool. http://apps1.eere.energy.gov/buildings/energyplus/.
[4] Intelmote2 and IMB 400 sensor board. http://www.memsic.com/.
[5] *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. Organization for Economic Cooperation and Development.
[6] Phoenix controls. www.phoenixcontrols.com/.
[7] TEP105: Low Power Listening. http://www.tinyos.net/tinyos-2.x/doc/html/tep105.html.
[8] Trueview people counter. http://www.cognimatics.com/.
[9] ASHRAE standard 55: Thermal environmental conditions for human occupancy. ASHRAE,Inc., 2004.
[10] ASHRAE standard 62.1: Ventilation for acceptable indoor air quality. ASHRAE,Inc., 2007.
[11] Y. Agarwal, B. Balaji, S. Dutta, R. Gupta, and T. Weng. Duty-cycling buildings aggressively: The next frontier in HVAC control. In *IPSN*, 2011.
[12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *CVIU*, 2008.
[13] V. Erickson and A. Cerpa. Occupancy based demand response HVAC control strategy. In *BuildSys*, 2010.
[14] V. Erickson, Y. Lin, A. Kamthe, R. Brahme, A. Cerpa, M. Sohn, and S. Narayanan. Energy efficient building environment control strategies using real-time occupancy measurements. In *BuildSys*, 2009.
[15] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa. OBSERVE: Occupancy-based system for efficient reduction of HVAC energy. In *IPSN'11*.
[16] W. Fisk, D. Faulkner, and D. Sullivan. Accuracy of CO2 sensors in commercial buildings: a pilot study. Technical report, LBNL, 2006.
[17] A. Kamthe, L. Jiang, M. Dudys, and A. Cerpa. SCOPES: Smart cameras object position estimation system. In *EWSN'09*.
[18] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900–I–903 vol.1, 2002.
[19] A. Logic. http://www.automatedlogic.com/.
[20] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
[21] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart

thermostat: using occupancy sensors to save energy in homes. In *SenSys*, 2010.

[22] D. Lymberopoulos, A. Bamis, and A. Savvides. The behaviorscope framework for enabling ambient assisted living. In *Personal and Ubiquitous Computing*, 2009.

[23] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *SenSys'04*.

[24] OpenCV. http://opencvlibrary.sourceforge.net/.

[25] J. Scott, A. J. B. Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar. Preheat: controlling home heating using occupancy prediction. In *UbiComp'11*.

[26] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2006.

[27] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

APPENDIX

A. MARKOV CHAIN MODEL

A Markov Chain (MC) framework is used to model the temporal occupancy dynamics of the building. States of the MC are the room occupancies of the building where the transitions among states occur with a probability dependent on time. This is described in detail in [15].

This framework allows the occupancy prediction $t + \Delta t$ in the future given the occupancy distribution at time t. This is done by multiplying the Δt times of the transition matrix. Each state of the MC is defined using a vector of building room occupancies (Figure 19). Let R represent the set of n rooms within a building. If we assume a maximum occupancy for each room in R, then we define $S = \{s_0, ..., s_m\}$ to be the set of all combinations of room occupancy where m is the total number of states. Thus, S represents all observable occupancy states that can be represented by a given the maximum occupancy constraints of rooms in R. As the number of rooms increases, the total number of observable states increases exponentially. In order to reduce the total state space, the MC is defined only using states observed during training.

Next we define transition probability matrix. Let the transition probability of moving from state j to i be represented by p_{ij} and X_t represent the occupancy state at time t. We calculate $p_{ij} = P(X_{t+1} = i | X_t = j)$ for each state in S. The time step of the MC represents 1 second, which is determined by the training data resolution. The transition probabilities are estimated by normalizing the transition 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.

Room occupancy varies depending on the time of the day and must be a component of the model. We incorporate time by defining multiple transition matrices that govern the state changes within different time windows, defining an inhomogeneous MC. Since our state space only considers the states observed in the training state rather than all possible states it is possible to have discontinuities at the slot boundaries. Assume data is partitioned into hourly transi-
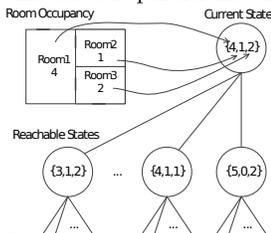

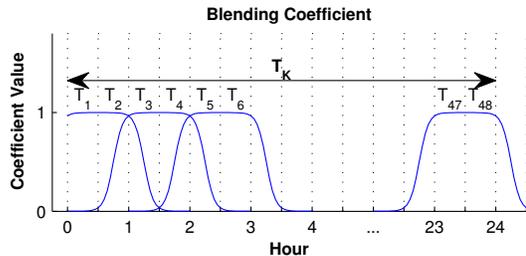**Figure 19: The occupancy state representation.**

Figure 20: Blending the transition matrices.

tion matrices and we are predicting occupancy for hours h and $h + 1$. Suppose we are in some state X and the hour changes from h to $h+1$. At this point the model switches to the hourly transition matrix for $h+1$. It is possible the transition matrix $h + 1$ has no probability for occupancy state X. If X does not occur in hour $h + 1$ of the training data, the transition probabilities for X cannot be calculated.

A.1 Blended Markov Chain

A blended Markov Chain (BMC) is a framework that avoids the discontinuities at the slot borders. This is done by ensuring that a transition probability for the current state always exists by blending each transition matrix into a single state space containing all the states from the training set. Let the day be partitioned into K parts. Thus, we have K transition matrices $T_1 \ldots T_K$ each with m states. These K transition matrices are then linearly combined to construct K blended transition matrices $\overline{T}_1, \ldots, \overline{T}_K$. The slot t blended transition matrix is then expressed as $\overline{T}_t = \sum_{s=1}^{K} \beta_{ts} T_s$ where $\beta_{t1}, \ldots, \beta_{tK}$ are positive and sum to one; this is done to ensure that $\overline{T}_t$ is a valid transition matrix. These coefficients are approximately 1 for close slots and quickly decrease to 0 for farther slots. This is done by defining the coefficients as

$$\beta_{ts} = \frac{\alpha(c_t - c_s)}{\sum_{s'=1}^{K} \alpha(c_t - c_{s'})} \tag{4}$$

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

$$\alpha(x) = \sigma\left(\frac{2a}{d}\left(x + \frac{d}{2}\right)\right) - \sigma\left(\frac{2a}{d}\left(x - \frac{d}{2}\right)\right) \qquad x \in \mathbb{R} \tag{5}$$

where $\sigma(x) = 1/(1 + e^{-x})$ (the sigmoid function), $a > 0$ is the slope at the slot boundaries and $d > 0$ is the slot width. The sigmoid is monotonically increasing where $\sigma(-x) = 1 - \sigma(x)$. Thus $\alpha(x)$ is positive and symmetric around its center. Figure 20 shows several slots. Using this construction, each state in the blended transition matrix $\overline{T}_s$ incorporates data from all slots, but heavily favors slot s. Our model utilizes $K = 48$ (half our slots) with slot width $d = 3$. We set a slope of $a = 10$ and choose the c_k to be center of the current hour. Figure 20 shows the blending coefficient for these parameters. These parameters create overlapping slot boundaries, increasing the number of preferred states available for transitions and minimize the probability of transitioning to a state completely outside the slot boundaries. States within a given hour are given higher weights are and states from adjacent half-hour slots are somewhat considered. States completely outside this time frame are given very low weight.