Improving Building Energy Efficiency by Kinect-based Occupancy Tracking and Mobility Detecting System

Haleimah Al Zeyoudi   Yanan Xiao   Chi-Kin Chau
Masdar Institute of Science and Technology P.O. Box 54224 Abu Dhabi, UAE
{hzeyoudi,yxiao,ckchau}@masdar.ac.ae

ABSTRACT

Nowadays, most building air conditioning systems still operate on a fixed schedule rather than real-time occupancy. In our study, we make an occupancy tracking software based on Kinect to reflect the number of people in a open lab. We then build a Markov Chain (MC) model after dividing the open area into 4 zones and calculating its occupancy respectively. When applying the real-time schedule of one week to a building model created with eQuest, we obtain a 22.1% energy reduction in space cooling.

1. INTRODUCTION

At the core of energy consumption in modern buildings is Heating Ventilation and Air Conditioning (HVAC) systems which are designed to operate at full capacity most of the time because it's often assumed a maximum occupancy, according to [3]. Although current HVAC systems are equipped with sensors, their management and control systems ignore the dynamic nature occupancy patterns in buildings. In addition, they are unable to proactively adjust to occupants' comfort levels. Understanding human mobility and occupancy patterns are key factors in successfully managing HVAC systems in buildings. The main contribution of our paper is to propose an energy-saving model based on occupancy patterns of human mobility in buildings. The most important features of the system are as follows: a) A real-time detection and tracking of human mobility in buildings based on Kinect. b) An occupancy prediction mechanism based on MC.

2. IMPLEMENTATION

2.1 Kinect-based Occupancy Counting Software

Our occupancy counting software is written using Microsoft Visual C# 2010, project WPF application in C# and XML. Both types of Kinect are used and tested to insure that it works properly, i.e. Xbox Kinect and Kinect for Windows. The software, should be run on windows 7 or above, after a successful installation of Kinect drivers.

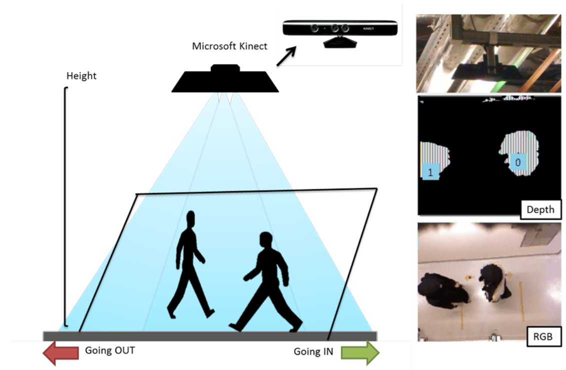

Figure 1: Architecture of human mobility tracking system

Once the software is started, it will do the self-diagnosis to check if a Kinect is connected to the computer properly. Then both the red, green and blue (RGB) and depth images are captured with the help of color sensor as well as infrared (IR) sensor, which comes with Kinect. The depth image as shown in Figure 1. For each depth image captured, a corresponding blue box with unique ID is drawn to detect the movement of multiple people at the same time. We implement both human posture tracking and skeletal tracking for a higher detection accuracy. Reference code of the latter can be found in MSDN library.

2.2 Testbed Setup

The environment we have is an open laboratory with no doors separating each professor's office. Therefore we divide this lab into 4 zones and deploy 8 Kinect sensors along with a cheap laptop running Windows at key areas of each zone, i.e. the conceptual entrance and exit. The in and out image is captured by Kinect and processed in Windows. All the in and out data is stored in laptops and synchronized via Dropbox. A selection of deployments is shown in Figure.

2.3 Markov Chain Prediction Model

We decide to use MC to build our prediction model after comparing results in [1,2]. We start by parsing in and out log files into occupancy files, and then inspect the distribution of occupancy data, i.e. number of people at a specific time slot in each zone. We adopt a time slot of 30min in the

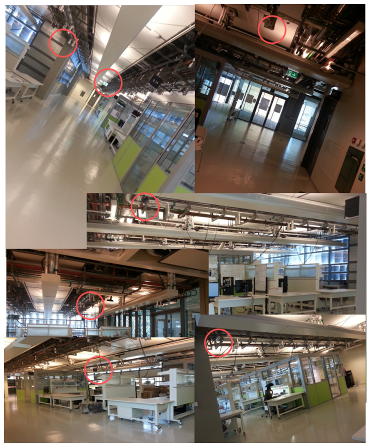

Figure 2: Kinect deployments in an open lab

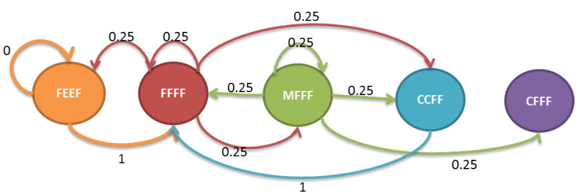

Figure 3: Simplified transitional matrix

end and scale the occupancy data down into 4 states: E for empty, F for few, M for medium and C for crowded. For each time slot, one and only one occupancy state is assigned with respect to the number of people in that zone. For example, when the number lies in between 8 and 14, then the state is marked M.

Judging from the high accuracy of our Kinect detection system, which we conclude by manually counting the captured videos and comparing with software-generated data, we use the occupancy data alone to train Markov model. Briefly speaking it is a 256-by-256 matrix due to the fact that we use a zone-based method to deal with this open lab, and each zone may be at one of four states when doing the permutation.

The matrix we get is extremely sparse because most of the states and the next state to which it will jump do not exist in reality. The simplified states and their transitional relations are shown in Figure 3. This transitional relation is then used to build a dynamic control system with which the ventilation and air conditioning systems are adjusted.

3. SIMULATION

We carry out the simulation by creating a building model and changing corresponding parameters with eQuest. The model is a two story office building with a total floor area of $232m^2$. Each floor is divided into four office zones with a floor area of $40m^2$ each. Windows are placed on the north, east and west walls of the building with overhangs having a projection factor (overhang depth/window height) of 0.6. Two doors are placed on the north and east sides of the building. Windows and doors are not specified on the south wall to minimize heat gain through radiation. The window to gross wall area is kept at 29%. The cooling system used

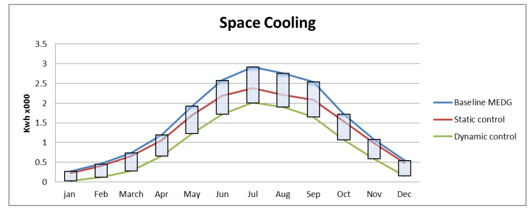

Figure 4: Energy consumption of 3 different control methods

is manually configured to match what is being used in our institute.

4. RESULTS

We adopt two control cases to see their effect on the model. For one thing, we use a fixed set-point for air conditioning system while applying a dynamic occupancy schedule generated from MC model. We do this because the initial building design guide assumes a maximum occupancy throughout the weekdays, which is far from being trustworthy according to our observations. For another, we apply the more authentic schedule alongside with a dynamic set-point that varies according to the occupancy states. The more populated a zone is, the lower set-point it will get. Our simulation result is shown in Figure 4. The optimum saving is 22.1% for month July.

5. CONCLUSION AND FUTURE WORK

We are making those heavily energy-consuming buildings at Masdar Institute more smart by developing wireless sensor networks to dynamically control the HVAC system. For now, we have a Kinect-based mobility tracking system to tell exactly there are how many people in each zone of an open lab. We build a model using Markov Chain to predict the likelihood of each zone's next state. And we have conducted several simulations to see the energy reduction. We plan to modify our MC model so that it will be more accurate since we have so many equal probabilities. We will implement the full-sized experiment on a field station once everything is settled.

6. REFERENCES

[1] Varick L. Erickson and Alberto E. Cerpa. Occupancy based demand response hvac control strategy. In *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pages 7–12, 2010.

[2] Varick L. Erickson, Yiqing Lin, Ankur Kamthe, et al. Energy efficient building environment control strategies using real-time occupancy measurements. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 19–24, 2009.

[3] Chuang Wang, Da Yan, and Yi Jiang. A novel approach for building occupancy simulation. *Building Simulation*, 4(2):149–167, 2011.