Study Space Occupancy Monitoring: Real-time Capacity Tracking and Reporting System

A Technical Report Submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Alby Alex

Fall 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Briana Morrison, Department of Computer Science

Study Space Occupancy Monitoring: Real-time Capacity Tracking and Reporting System

CS4991 Capstone Report, 2024

Alby Alex
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
tcv6rb@virginia.edu

ABSTRACT
At times during the semester and exam periods, it becomes difficult to find free space at the numerous study spaces across Grounds at the University of Virginia. To resolve this issue, I propose creating a web application that tracks how full each study space is and publishes that information in real time for students to view. This application would make use of infrared sensors placed at the entrances and exits of each building. This uploaded information would then be displayed on a web application that provides a percentage describing how full each space is. The application would allow students to make more informed decisions about where to go. As a result, students will be more dispersed, and individual spaces will not be as crowded as usual. If successful, this concept could be extended to dining halls and restaurants on Grounds, as well.

1. INTRODUCTION

Imagine that you are a student at the University of Virginia. It is the last Saturday before your first final exam. You wake up at 9:00 AM and decide to spend the whole day studying. Your first choice of study spot is Clemmons Library, and you make your way there. When you reach it, you find that there is no comfortable space anywhere in the building to sit and study. Next, you try to find space at Alderman Library, just to fail again. Finally, you head to Brown Library and find a comfortable space. However, it is now 10:00 AM, and you have lost much of the motivation you had at the start of the day.

This scenario could have been avoided with advance knowledge of which spots were close to capacity. The Study Space Occupancy Monitoring system will provide this advance knowledge to enable students to maximize their study time. It will also help in scenarios where groups of students need to know the locations of available collaborative spaces.

2. RELATED WORKS

Automated building occupancy monitoring has been consistently studied to increase energy efficiency in buildings. Zou, et al. (2017) proposed that this be achieved through a proprietary system called WinOSS, which generates occupancy information from the information transferred by devices connected to a particular WiFi network. My initial focus for the project was to utilize this described technology for occupancy monitoring, but it has some flaws that held me back. First, it presumes that every person entering a building has a device that will connect to the building's WiFi network. While this may be true in an office setting, it is less likely to always be true in college. Second, it may undermine the privacy of the people that enter the building because of the unique device identifiers that it collects.

For these reasons, I chose to utilize the methodology used by Mikkilineni, et al. (2019) in their study. Their solution utilizes thermal imagers with low-cost optical enhancements to count occupants. This method avoids the privacy concerns of the alternative, as any recorded information will consist solely of heat signatures. Previous iterations of this method were unreliable in certain scenarios, but these concerns can be addressed using image processing and computer vision to ensure an accurate count.

3. PROPOSED DESIGN

The core function of this project is accomplished through the interaction of two main components: the sensor system array for measuring current occupant counts in buildings; and the web service displaying the current occupancy status of each building. The current design of the system makes it easy to expand the network if a new study area is added to Grounds later on.

3.1. Sensor Array

The sensor array used in this project will utilize the designs provided by Mikkilineni, et al., in their 2019 publication. They outline how the individual sensors are designed, how they should be arranged in each space, how to calibrate the onboard counting algorithm, and how to communicate with the sensors. Each of these steps will be discussed in depth in the following sections.

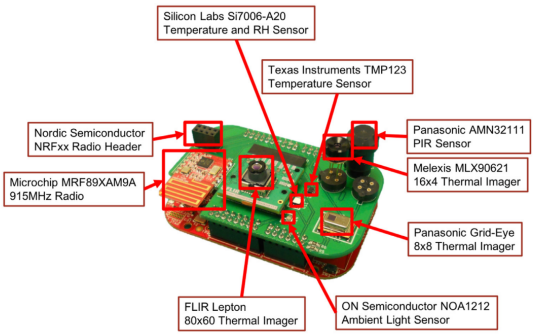

Figure 1. Detailed view of a sensor node. Source: Mikkilneni, et al. (2019).

3.1.1 Individual Sensor Node Design

As shown in Figure 1, each sensor node will consist of a Freescale/NXP KL46Z256 board with a multitude of different sensors, including multiple thermal imagers, a passive infrared sensor (PIR), and an ambient light sensor. The specific board type was chosen for its flexibility in programming and its cost effectiveness. The array of sensors on each node was chosen to provide the best cost-efficiency per node. The sensor also contains a radio to communicate with a central node that tallies the changes in occupation count. This node will be placed within a protective 3D-printed housing when it is put into action.

3.1.2. Sensor Arrangement

The reference paper describes a more precise methodology for the placement of sensors within the rooms under observation. However, this implementation is more precise than necessary for this project as it attempts to maintain a count of occupants in each zone of a building. In the context of the proposed project's application, counts of people are in the building or in each study area would be adequate. As such, the sensor nodes should be placed at all the entrances of the libraries and the study areas.

3.1.3. Occupancy Monitoring Algorithm

With the nodes placed in the appropriate locations, the next step is to determine how the nodes calculate the occupancy of any given room. As described by Mikkilineni, et al., this algorithm makes use of basic image processing techniques including thresholding, region growing, and blob detection. The sensor nodes will be placed on doorways looking straight down.

In essence, the sensor data can be collected and analyzed based on the temperature data of each pixel in the image. A rolling average of temperature data for 20 frames is recorded for each pixel. If the temperature of the current

reading is higher than that rolling average, it can be assumed that there is a person present on that pixel because of their natural body heat. After this, a blob detection algorithm connects contiguous pixels with detected heat and tags each with a label.

Figure 2. This image shows how the field of view of the sensor node can be divided up. Source: Mikkilneni, et al. (2019).

The locations of these blobs are then placed into three different bins based on the image shown in Figure 2. The blobs are tracked between frames using the labels discussed earlier. If a blob goes missing between two frames, the occupancy count is increased or decreased based on which zone the blob was last seen in. If a new blob is found, then it is added to the list of currently tracked blobs.

3.1.4. Communication of Occupancy Counts

Each sensor node has two modes, control or sense. For a set of sensor nodes in a single building, there needs to be one control node. Whenever a sensor node detects an occupancy change, it communicates that change to the central node. The central node then adds that information to the internal count of the space's occupancy that it maintains. This solution avoids making the system reliant on local WiFi access. The data from the central node can be read by a computer over a USB connection.

3.2. Web Application

The data from the different sensor arrays will be communicated to a central server that keeps track of the occupancies of each study area and displays this information for public consumption.

3.2.1 Web Server Configuration

This web application will have two parts: a private REST API that collects updates from the sensor arrays; and the public-facing website. Both parts will operate on the Django webserver framework because of its simplicity and robust community support. The private REST API will require authentication from the computers to upload values to avoid any incorrect values. Each location will start with an initial occupancy count of 0, and it will be updated whenever the sensor array of each area detects a change. Each time someone visits the public-facing website, the current value of the occupancy for the chosen location will be used to provide an estimate. This webserver will be deployed on the AWS Cloud to ensure uptime and robust load management.

3.2.2 Public-Facing Website

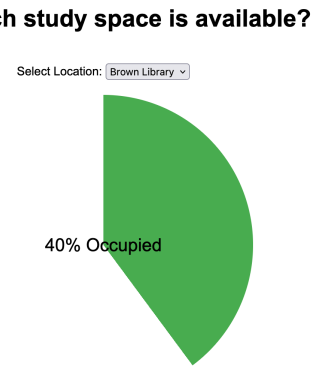

Figure 3. This image shows a Mockup of what the Student-Facing Side of the Website will resemble.

The website will resemble the site shown in Figure 3. This percentage will be based on the current occupancy count stored on the webserver and a pre-calculated value that measures how many people each study area

can hold. Users can easily choose between study areas and understand where the current least occupied positions are. This website will be compatible with both computers and mobile devices.

4. ANTICIPATED RESULTS

Ideally, this system should make it easier for people to plan their study locations in an efficient manner. It should boost productivity and motivation for studying and cut down on time wasted trying to find a space. Students will be better distributed around Grounds, helping to ease the load for custodial staff at high traffic areas. Furthermore, it should help to move people around Grounds to learn about new areas. Students will gain an appreciation for all that Grounds has to offer.

The system should be periodically evaluated for its accuracy through a manual count of the people in each study area. The results of this evaluation should determine if any tweaks to the algorithm are necessary.

5. CONCLUSION

The implementation of this project will benefit students in a multitude of ways. If every student takes advantage of this service, locations on Grounds will become significantly less crowded and more people will be able to use them. University administration can also use the data provided by this project to figure out where students want to spend their time and allocate resources to support students based on this information. After an initial calibration period, the service will become fully automated with minimal maintenance costs. In addition, new construction on Grounds can be easily accommodated by the application with the installation of a new sensor array. People want to enjoy the amenities that they have access to at UVA, and this application would make that process seamless.

6. FUTURE WORK

The implementation of this proposal should take a gradual approach. The sensor array should be installed in one location and monitored for a period to ensure that the provided count is accurate. Once the accuracy is confirmed, the web application to collect and display the data should be created and tested thoroughly. Finally, the sensor array should be installed at every study location and the application should be made available to the student population. If it proves to be successful, this project can be expanded to cover the dining halls and restaurants on Grounds as well.

REFERENCES

Aravind K. Mikkilineni, Jin Dong, Teja Kuruganti, and David Fugate. 2019. A novel occupancy detection solution using low-power IR-FPA based wireless occupancy sensor. *Energy and Buildings* 192 (June 2019), 63–74. DOI:http://dx.doi.org/10.1016/j.enbuild.2019.03.022

Han Zou, Hao Jiang, Jianfei Yang, Lihua Xie, and Costas Spanos. 2017. Non-intrusive occupancy sensing in commercial buildings. *Energy and Buildings* 154 (November 2017), 633–643. DOI:http://dx.doi.org/10.1016/j.enbuild.2017.08.045