Monitoring Activity in the Math and Science Center

Ryan Strauss
rystrauss@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

This project investigates patterns in the use of Davidson College's Math and Science Center (MSC). The MSC is a great resource for STEM students in need of extra help, but it can often be hard to get that help during busy hours. Using a Raspberry Pi, data was collected on the level of activity in the MSC over the course of a week. The results provide insight into the best times to visit and avoid the MSC.

1 Introduction

One of the most popular services provided by Davidson College's library is the Math and Science Center (MSC). This center is an important resource for students in STEM classes. It provides free peer tutoring in STEM coursework by students who have already completed such work.

Unfortunately, the MSC is often overcrowded, and it is not always guaranteed that a tutor will be available. This project attempts to provide insight into this problem by collecting data on MSC usage over the course of a week and extracting patterns from that data. Data collection is accomplished with a Raspberry Pi, various sensors, and computer vision.

2 Background

The MSC is a place for student tutors to help peers with their coursework in the maths and sciences. Located in the E.H. Little Library, this center is an essential resource for many students when they are struggling in their classes. The trained and highly qualified peer tutors in the MSC support students in all areas of quantitative or scientific reasoning. The center also provides a gathering place where students can work collaboratively and a quiet study space for students working and studying individually.[1] Assistance from tutors who have completed the same coursework can make all the difference for students as they learn new concepts.

However, the MSC is known to have certain problems with staffing. There are often times when not enough tutors are on shift to meet the overwhelming demand, and many students end up not being able to receive help. Long wait times turn potential tutees away, which eliminates the benefit of having the MSC in the first place.

[1] https://www.davidson.edu/offices/ctl/students/math-science-and-economics-center

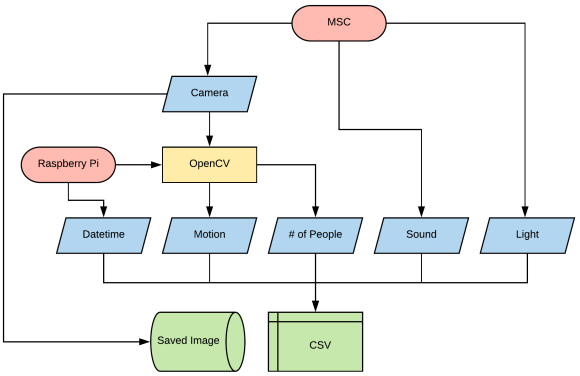

Figure 1: Flowchart illustrating data collection and processing.

For students, it can be hard to know what times the MSC will be busiest or quietest. With an understanding of the trends in MSC activity, these problems can be more easily solved, and that is what this project aims to do.

3 Solution

Understanding the Problem

In order to provide a solution to a problem, that problem must first be well-defined. At a high level, this problem can be defined as needing to determine how 'active' the MSC is at a given time. How does one define 'active,' though? What does it mean for the MSC to be 'busy' or not?

An obvious method for determining the activity is to count the number of people in the MSC. Too many people is, after all, the problem being addressed in this project. There are also more indirect measurements of activity in the center. These include sound level, whether or not there is motion, and how many lights are on in the room.

The combination of these factors can help paint a picture of how busy the MSC is at a given time. So, this project provides insight into the problem by collecting data on these factors over the course of a week, with the hope that any patterns in MSC activity will be revealed.

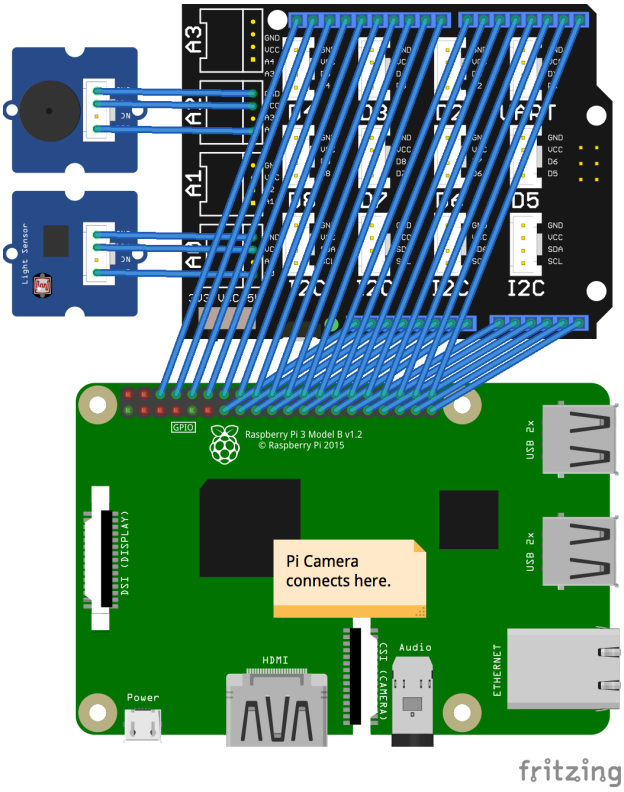

Figure 2: Diagram of listening station components. Sensors are Grove sound sensor (top) and Grove light sensor (bottom). Pi Camera is not shown.

Collecting Data

This project needed to unobtrusively collect the previously mentioned information for a week. This was accomplished with a Raspberry Pi 3. The Raspberry Pi is small enough to be easily disguised for concealed data collection and has the necessary GPIO connections for compatibility with requisite sensors. Using Python, the data can then be processed and aggregated as shown in Figure 1. Data was collected between 5/1/18 and 5/8/18.

Sensors As previously mentioned, there are four features that the Pi needed collect data on: sound, light, motion, and number of people. Sound and light data are both collected using the respective Grove sensors from Seeed.[2] These sensors connect to the Pi through a GrovePi+ (see Figure 2). Motion and people data both originate from camera input. A Pi Camera is connected to the Raspberry Pi through the dedicated port.

Processing Camera Data The Pi Camera provides a video feed of the MSC, however that feed needs to be transformed into the desired features (i.e. whether or not there is motion, the number of people in the room). This is accomplished with OpenCV (Open Source Computer Vision Library) (Itseez 2015). This library is aimed at real-time computer vision, making this project a perfect application.

[2] `https://www.seeedstudio.com`

Min	Max	θ	Resolution	FPS
1500	40000	2	1296×730	16

Figure 3: Values of data collection parameters. θ is the threshold for difference detection.

There are two components to the OpenCV processing used in this project, and they both implement motion detection (Rosebrock 2015). First, OpenCV simply analyzes the video to determine whether or not there is motion in the frame at a given point (i.e. when data is collected). It does this by keeping track of an average 'background frame.' This background frame represents the parts of the frame that never move. Then, every new frame is compared to the background frame, and if there is any difference between the two, the program determines that there is motion in the frame. Differences are only recognized if greater than a predefined threshold, θ, so that noise in the image is ignored.

Finally, the Pi must be able to count the number of people in the room. This process happens in simultaneity with the previously discussed motion detection. When difference detection is executed on every frame, it does so such that any differences are grouped into *contours*.

If a frame is a graph F where each pixel is a vertex, a *contour* is a spanning tree T of a subgraph G of F such that

$$\forall v \in V(T) : \Delta v > \theta \tag{1}$$

and

$$\min \leq |V(T)| \leq \max \tag{2}$$

where *min* and *max* are predefined values. This definition proves useful, as is allows large clusters of changing pixels to be interpreted as single 'objects.' Given that, by and large, humans are the only moving objects in the MSC, this effectively becomes a method for tracking people in the room.

There are a few key problems with this method, though. How is the computer to know whether or not a contour is a person? As previously mentioned, humans are generally the only things that will be moving in the frame, however this cannot be guaranteed. A person's head and arm could be accidentally counted as individual objects or a small change in the background could trigger a contour. The parameters θ, *min*, and *max* are intended to address these problems. By setting proper values for these parameters (see Figure 3), the program will only track contours with a specific area and only if there is a significant enough difference from the background. While this method is still hardly perfect, it is certainly better than any feasible alternative.[3]

Data Aggregation and Storage In order to discover trends in the data over time, data is collected at regular intervals. At the beginning of every 24-hour period, a new `DataFrame` (McKinney 2014) is created to hold the data for that interval. Then, at the beginning of every minute, the following features are captured as the given datatypes:

[3] The next best alternative would be single-frame object detection via haarcascades. However, there is too much uncertainty in the arrangement of the subjects for this method to be viable.

- Timestamp: `datetime` object
- Light: `int`, higher value means more light
- Sound: `int`, higher value means louder sound
- Motion: `bool`
- Contour Average: `float`, the average number of contours per frame for the given minute
- Image Name: `str`

This data is then appended to the current `DataFrame` as an entry. The video frame that was analyzed is then saved as an image with bounding boxes drawn around all contours that were found in the frame (the 'Image Name' feature refers to this image's filename).

After 1440 entries have been collected (one days worth), the current `DataFrame` is saved to a CSV file, then freed from memory. The entire process then repeats. This is done to reduce the memory requirements of the program and to ensure data is saved reasonably frequently.

Figure 4: Raspberry Pi as positioned during data collection (black and red 'Davidson' binder on top-left of the bookshelf).

Collection Device Disguise

In order to disguise the Raspberry Pi and sensors into the environment, they are enclosed within a three-ring binder. During the data collection period, the binder sat on top of a bookshelf in the MSC, which effectively camouflaged the device (see Figure 4).

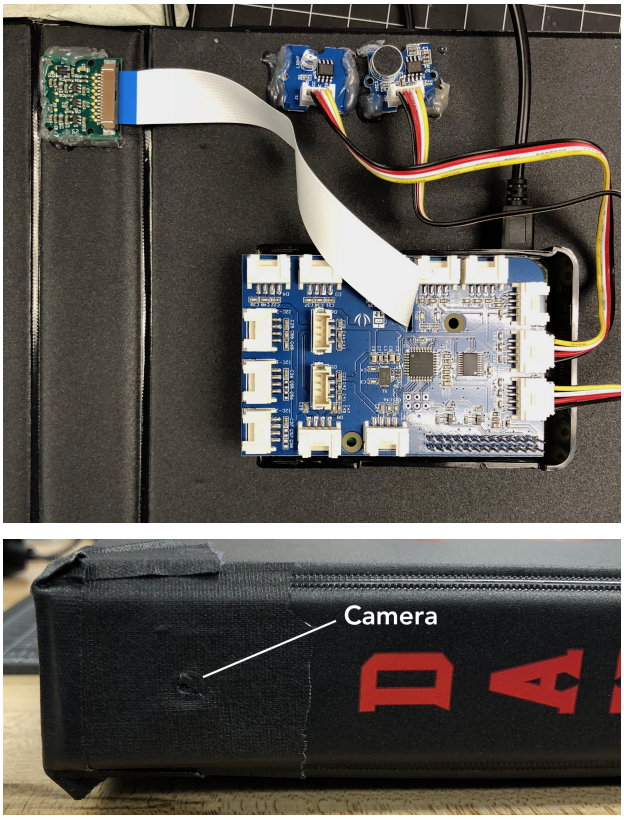

Figure 5: Two views of the binder disguise. Top image shows Raspberry Pi and sensors positioned inside the binder. Bottom image shows where camera is hidden.

The Raspberry Pi, light sensor, and sound sensor are simply glued to the inside of the binder so that they are concealed. The Pi Camera rests in a small hole in the spine of the binder so that it can 'see' the room (see Figure 5). Power is provided via an extension cord that runs down the back of the bookshelf.

4 Results

The basic question that this project attempted to answer was when is the MSC most busy? As seen in Figure 6, the late afternoon and evening is when the average number of contours is highest (represented by the green lines). In contrast, it is also apparent that most of the daytime hours (and middle of the night) see extremely low activity.

Unsurprisingly, the amount of light in the room appears to be directly related to the time of day (also shown in Figure 6). What is surprising, though, is that the more busy times tend to be at most no louder than the inactive times (notice the downward direction of the green lines on the right side of Figure 6, and the upward direction of the reddish lines).

Figure 7 reveals more about the relationship between light, sound, and activity. There appears to be very low correlation between light and sound (coefficient of determina-

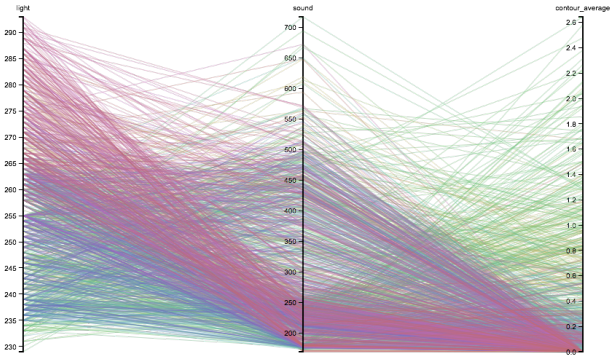

Figure 6: Parallel coordinates plot for light, sound, and contour average from May 1-2 (a Tuesday and Wednesday). Time of day is represented by color, with reddish-purple being noon and dark blue being midnight. Outliers are excluded from this visualization.

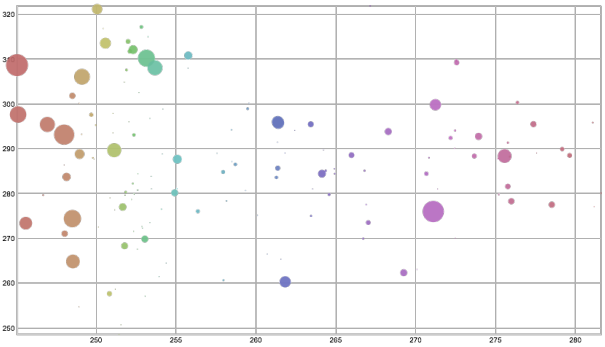

Figure 7: Scatter plot where each point represents the averaged values for a given hour during the collection period. Light is on the x-axis and sound is on the y-axis. Size represents contour average and color represents time of day (red being day and purple being night). Outliers are excluded from this visualization.

tion is close to zero). Also, the size of the points seems to be most dependent on the time of day, which corroborates the conclusions drawn from Figure 6.

Also, as shown in Figure 8, Tuesday through Thursday are the busiest weekdays.

5 Conclusions

In order for the MSC to be an effective resource, it is necessary that students are able to find an available tutor when they visit. During its busiest times, though, this is not always the case. By collecting data on MSC activity with a Raspberry Pi, this project discovered that the center is busiest during evenings. Based on this information, it can be determined that students seeking help will most likely be successful if they visit the MSC between 4:00 and 6:00 (the first shift for many of the tutors) and not on Tuesday, Wednesday, or Thursday. The results also show that the amounts of light and sound in the MSC are not necessarily determinants of

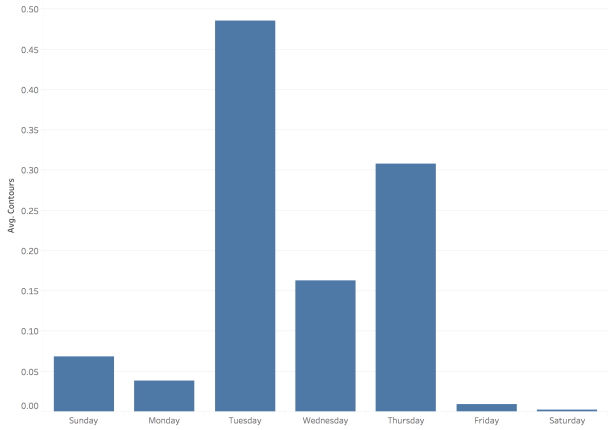

Figure 8: Bar chart showing the average number of contours per minute for each weekday.

Figure 9: Image of MSC captured at 5:36 PM on May 1st. The green box in the bottom right of the image highlights an object that OpenCV recognized as in motion. It is evident that only one of the five people in the room was detected.

activity.

It is important to point out a certain degree of error in the data collection, though. Specifically, the processing of the camera data did not always accurately reflect the number of people in the room. More subtle movements would often go undetected by the computer vision algorithm, not to mention people who were sitting still (see Figure 9).

Nevertheless, this project accomplished its goal of learning more about patterns in the usage of Davidson College's Math and Science Center.

References

Itseez. 2015. Open source computer vision library. `https://github.com/itseez/opencv`.

McKinney, W. 2014. Pandas, python data analysis library. 2015. *Reference Source*.

Rosebrock, A. 2015. Home surveillance and motion detection with the raspberry pi, python, opencv, and dropbox. `https://bit.ly/2HUIid2`. Retrieved on April 2, 2018.