Low-Overhead Occupancy Tracking

Hannah LaTourette
Master's Project, Spring 2021
Dr. Koushik Kar, ECSE

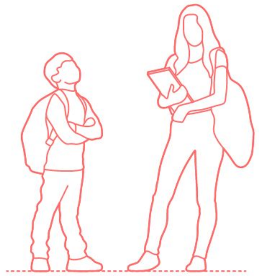
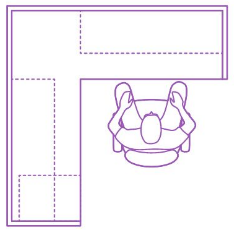
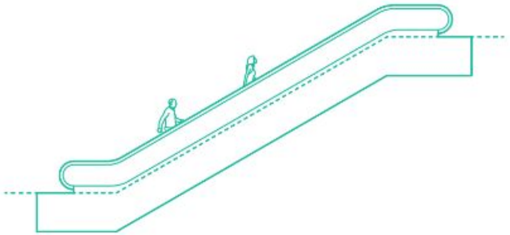
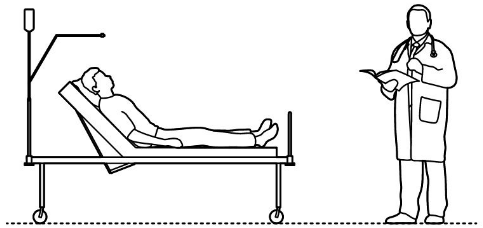
Overview

- Motivation and use cases
- System requirements
- Sensor specifications
- Algorithm design
- Software model
- Demonstration & evaluation
- Next steps

Problem Definition

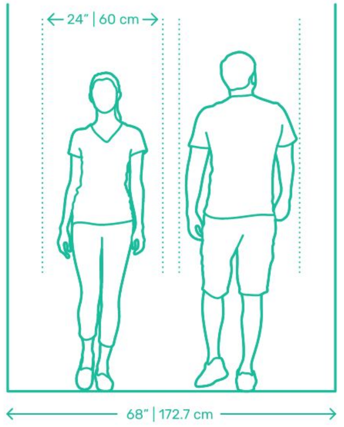
Use Cases

- **Universities**: attendance, real lecture sizes, use of on-campus resources
- **Large office spaces**: lighting and HVAC, emergency planning
- **Shopping malls**: traffic patterns, rent prices, or food cart locations
- **Intrusion detection**: alert when person is detected during the night, etc.
- **Social distancing**: limit people in stores without repeated counting

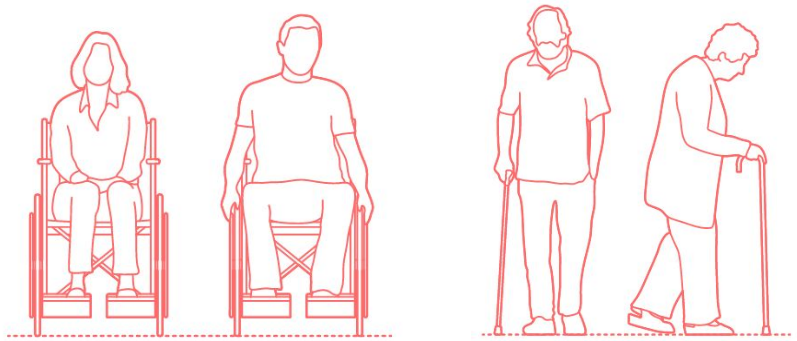
User Needs (Long-Term)

Need	Mall	University
Detection of multiple people walking together	<= 6 people	<=2 people
Detection of people walking through any part of corridor	<= 8 meters	<= 1.7 meters
Detection of people entering and exiting simultaneously	<= 3 per direction	N/A
Sensors which will last months to years	>= 1 year	>= 4 months or >= 8 months



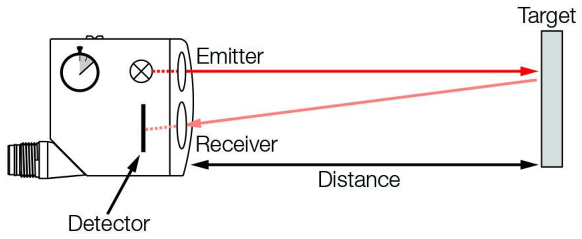
Additional considerations

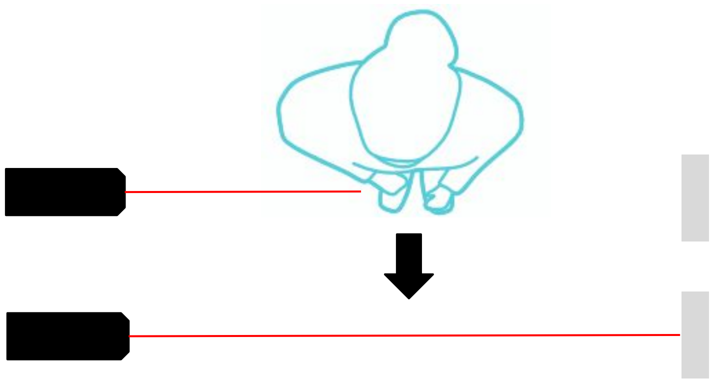
- Non-disruptive to those counted
- Not a privacy concern for those counted (issue for RF or cameras)
- Accessible (functional for people with wheelchairs, canes, etc.)



Solutions

Initial solution | Thru-beam





- If no object obstructs the path, the beam is reflected back to the sensor
- If an object obstructs the path, the beam will not be reflected back
- With two thru-beams, a person’s walking direction can be determined
- *Accessibility*: polarized sensors emit filtered light and expect it back, so they are not fazed by metal objects

Chosen sensor | Thru-beam

Banner Engineering Corp. S18-2VNLP-2M

Polarized retroreflective thru-beam sensor

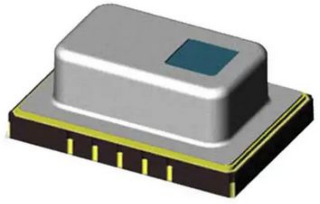
Sensing distance: 6 m

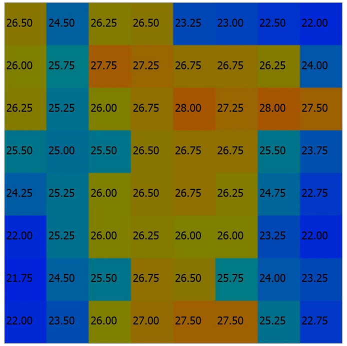
Power supply: 10-20 V

Communication: I2C

Response time: 1.5 ms

Chosen sensor | Coarse IR camera





26.50	24.50	26.25	26.50	23.25	23.00	22.50	22.00
26.00	25.75	27.75	27.25	26.75	26.75	26.25	24.00
26.25	25.25	26.00	26.75	28.00	27.25	28.00	27.50
25.50	25.00	25.50	26.50	26.75	26.75	25.50	23.75
24.25	25.25	26.00	26.50	26.75	26.25	24.75	22.75
22.00	25.25	26.00	26.25	26.00	26.00	23.25	22.00
21.75	24.50	25.50	26.75	26.50	25.75	24.00	23.25
22.00	23.50	26.00	27.00	27.50	27.50	25.25	22.75

Panasonic Grid-EYE AMG8854

64-pixel IR camera

Sensing distance: 7 m

Power supply: 5 V

Communication: I2C

Frame rate: 10 f/sec

Grid-EYE Development Board

Panasonic Grid-EYE AMG8854 EVAL

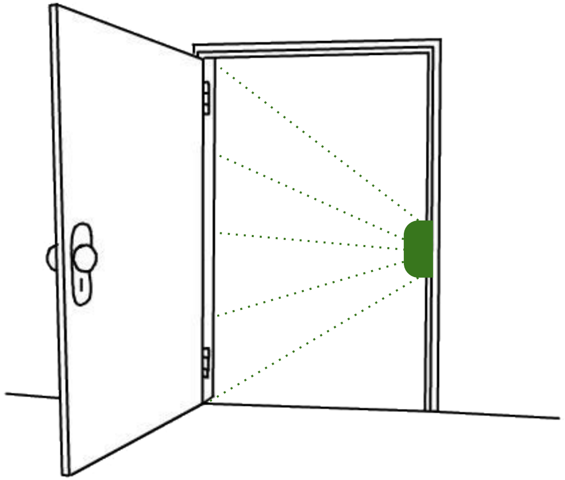
64-pixel IR camera w/ development board

- ATSAMD21G Microchip
- Bluetooth module
- Arduino header compatibility
- JTAG port

Due to time and resource constraints introduced by COVID-19, this board was used to develop a preliminary model.

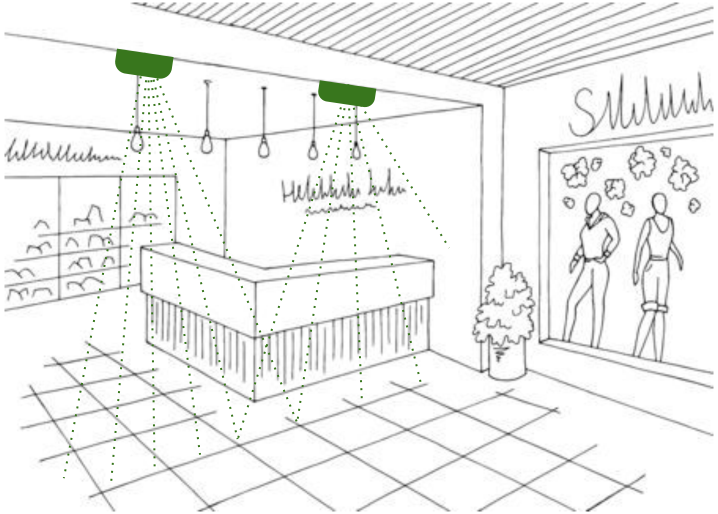
Project Basics

Physical setup



Spring 2020

Single sensor placed on side of doorway, only suitable to count one person at a time



Spring 2021

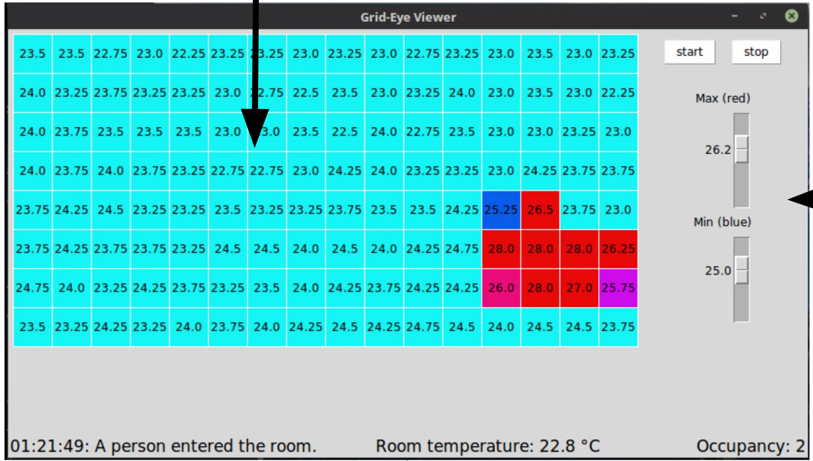
Multiple sensors placed on top or doorway or corridor, can count multiple entries / exits at once

Application overview

Desktop program written in Python using:

- TKinter
- PySerial
- NumPy
- GridEyeKit *(provided by Panasonic)*
- EvalKit *(provided by A. Hoch)*

Live color-coded grid showing current temperature values from sensors



23.5	23.5	22.75	23.0	22.25	23.25	3.25	23.0	23.25	23.0	22.75	23.25	23.0	23.5	23.0	23.25
24.0	23.25	23.75	23.25	23.25	23.0	2.75	22.5	23.5	23.0	23.25	24.0	23.0	23.5	23.0	22.25
24.0	23.75	23.5	23.5	23.5	23.0	3.0	23.5	22.5	24.0	22.75	23.5	23.0	23.0	23.25	23.0
24.0	23.75	24.0	23.75	23.25	22.75	22.75	23.0	24.25	24.0	23.25	23.25	23.0	24.25	23.75	23.75
23.75	24.25	24.5	23.25	23.25	23.5	23.25	23.25	23.75	23.5	23.5	24.25	25.25	26.5	23.75	23.0
23.75	24.25	23.75	23.75	23.25	24.5	24.5	24.0	24.5	24.0	24.25	24.75	28.0	28.0	28.0	26.25
24.75	24.0	23.25	24.25	23.75	23.25	23.5	24.0	24.25	23.75	24.25	24.25	26.0	28.0	27.0	25.75
23.5	23.25	24.25	23.25	24.0	23.75	24.0	24.25	24.5	24.25	24.75	24.5	24.0	24.5	24.5	23.75

01:21:49: A person entered the room. Room temperature: 22.8 °C Occupancy: 2

Adjustable scrollbars to customize cutoffs for blue and red color displays

Dashboard showing most recent event, room temp, and current occupancy

Important cases to handle

- Many people entering and exiting at once
- Lingering in the doorway to chat
- Walking parallel to doorway
- Child-height tracking

Algorithm Features

Counting occupancy

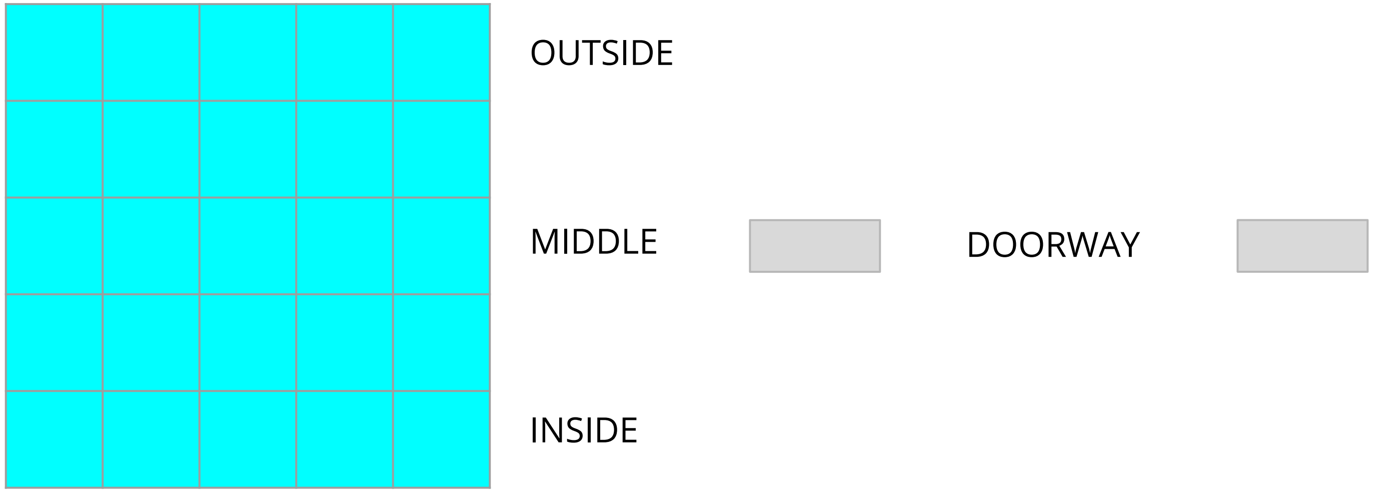
Clearing correct flags

Accounting for overlapping columns

Counting occupancy

Threshold crossing

If a person passes through a doorway into a room, they will pass from the outside, through the middle, to the inside:

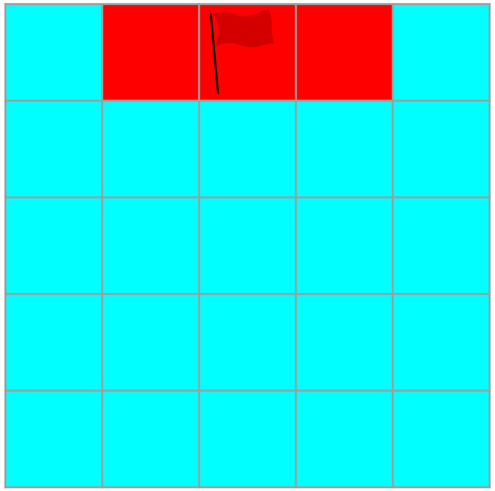


CURRENT OCCUPANCY: 0

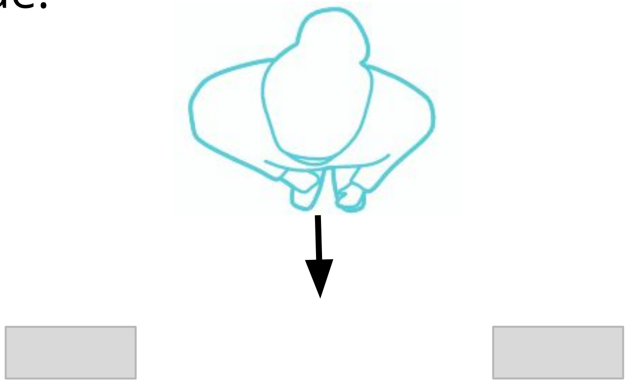
Counting occupancy

Threshold crossing

If a person passes through a doorway into a room, they will pass from the outside, through the middle, to the inside:



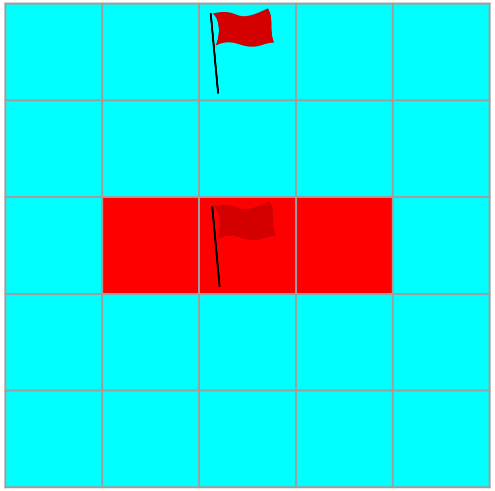
CURRENT OCCUPANCY: 0

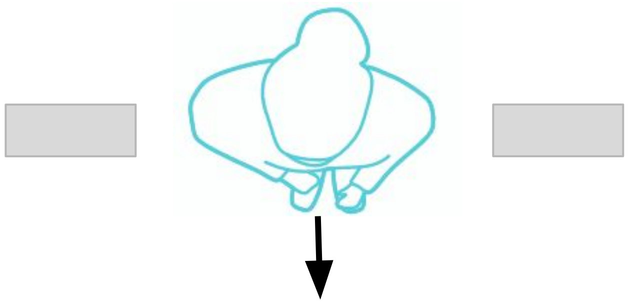


Counting occupancy

Threshold crossing

If a person passes through a doorway into a room, they will pass from the outside, through the middle, to the inside:



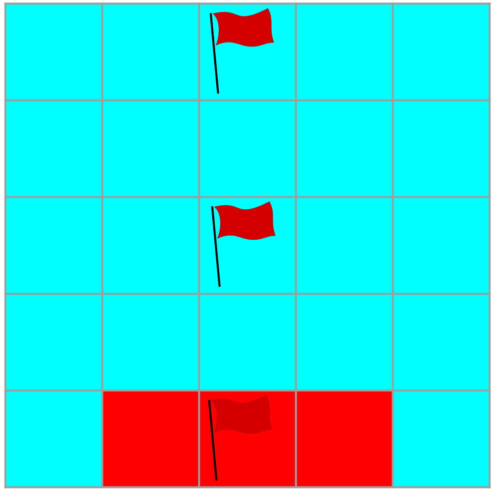


CURRENT OCCUPANCY: 0

Counting occupancy

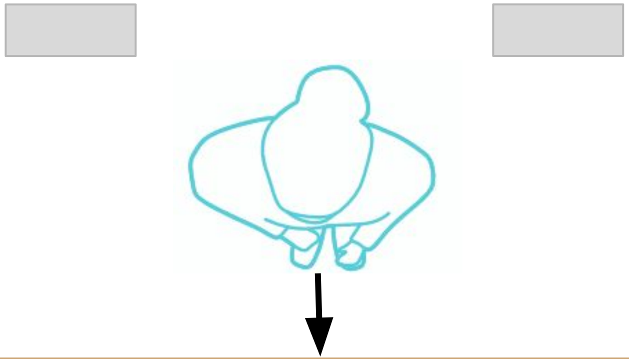
Threshold crossing

If a person passes through a doorway into a room, they will pass from the outside, through the middle, to the inside:



CURRENT OCCUPANCY: 0

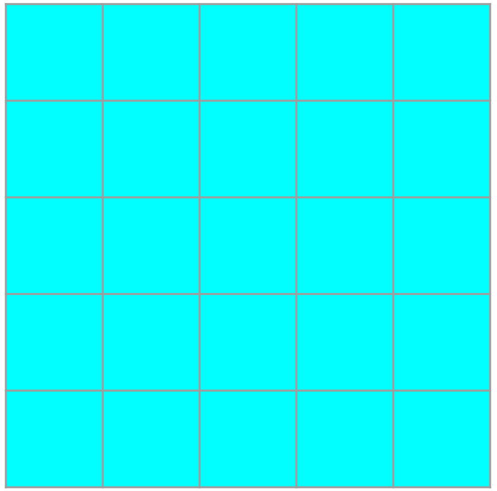
If the outside, center, and inside flags are all set, a person must have entered the room. Clear all flags and increment current occupancy by 1.



Counting occupancy

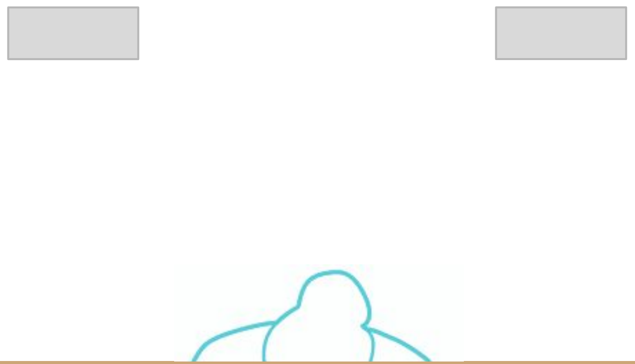
Threshold crossing

If a person passes through a doorway into a room, they will pass from the outside, through the middle, to the inside:



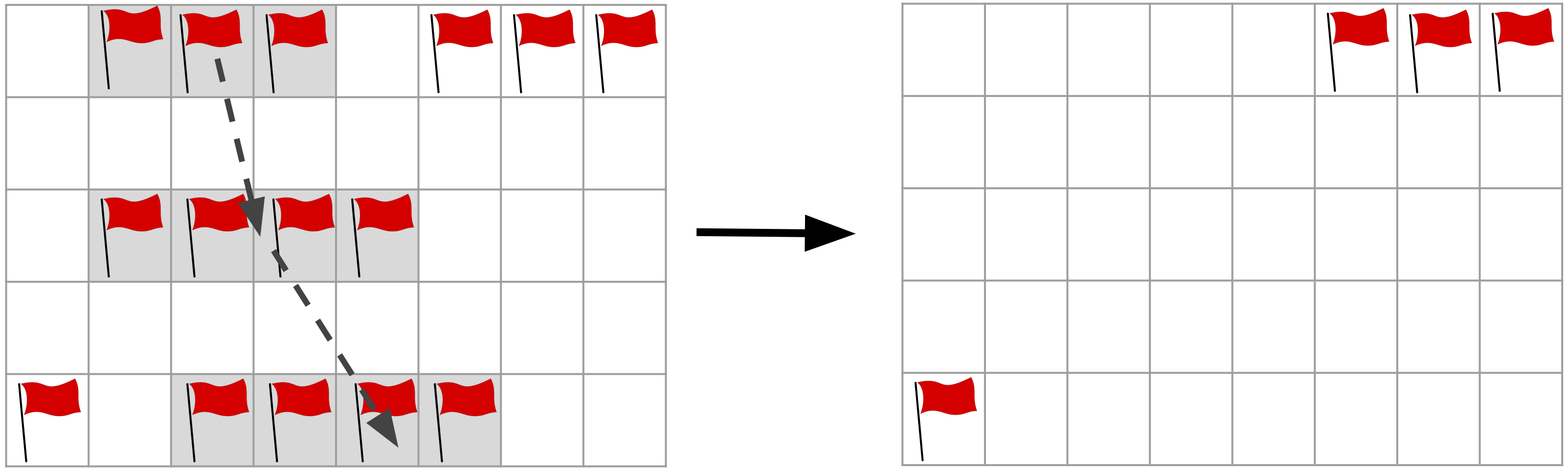
CURRENT OCCUPANCY: 1

If the outside, center, and inside flags are all set, a person must have entered the room. Clear all flags and increment current occupancy by 1.



Clearing flags

A person may set off heat flags in various columns during the walk across the threshold by entering at an angle, stumbling, avoiding others, etc.



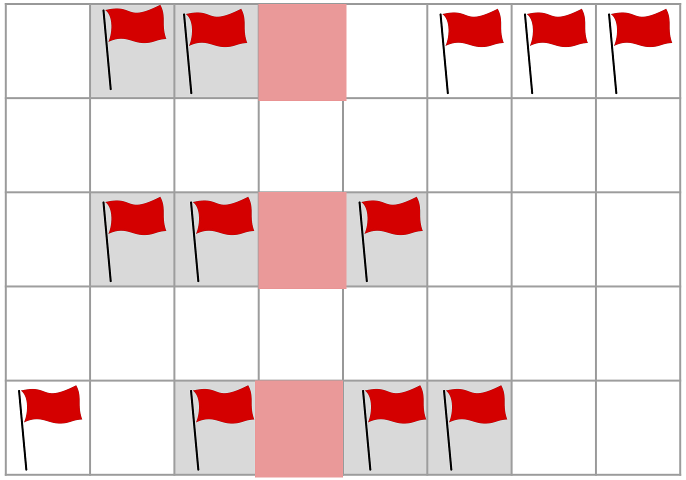
Person A’s path set off a variety of flags in different columns

Goal: identify the clusters of flags from Person A and clear them all

Clearing flags

Clustering

Solution: Record the column which sets its inside, middle, and outside flags first. Clear flags on either side of these flags until you hit a cell without a flag.

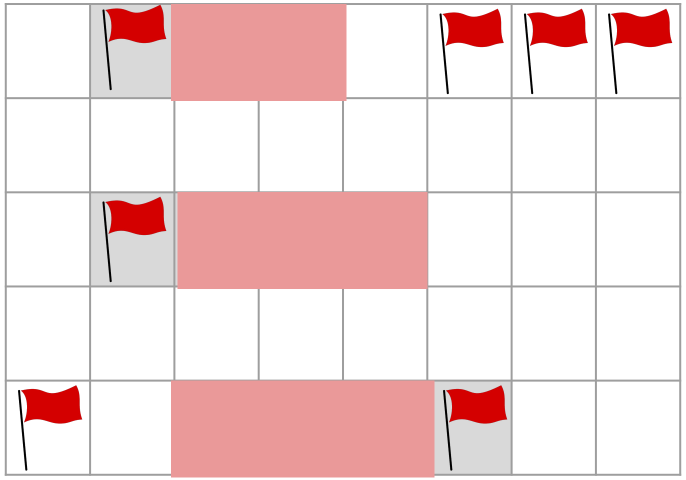


Person A's path set off a variety of flags in different columns

Clearing flags

Clustering

Solution: Record the column which sets its inside, middle, and outside flags first. Clear flags on either side of these flags until you hit a cell without a flag.

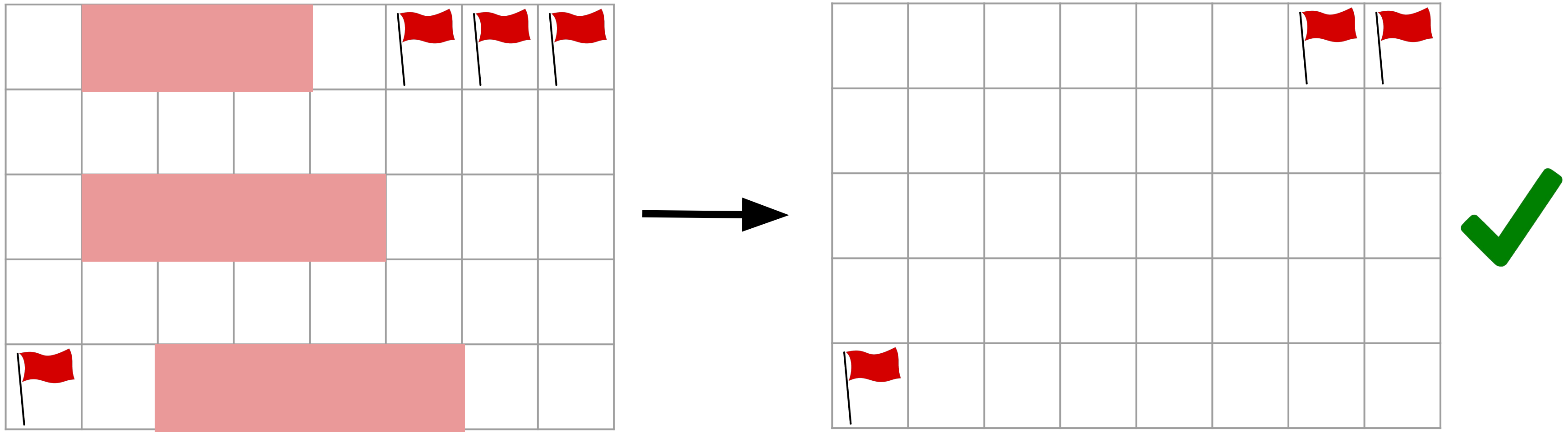


Person A's path set off a variety of flags in different columns

Clearing flags

Clustering

Solution: Record the column which sets its inside, middle, and outside flags first. Clear flags on either side of these flags until you hit a cell without a flag.



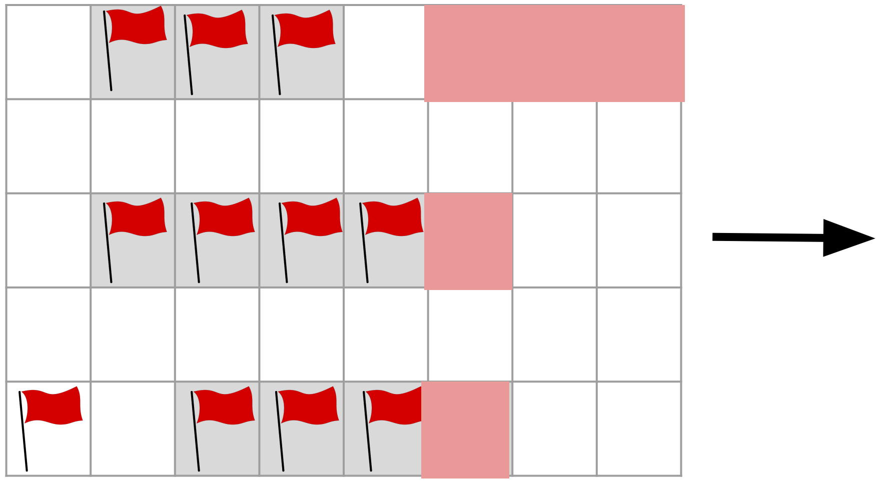
Person A's path set off a variety of flags in different columns

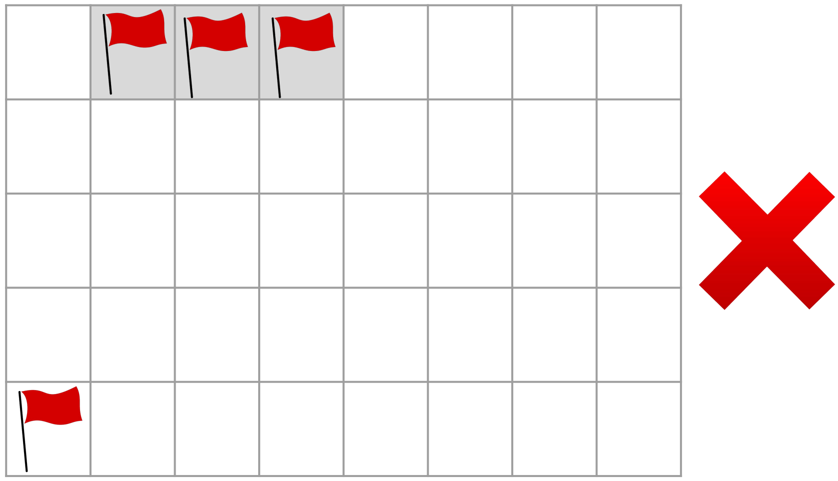
Correct flags cleared!

Clearing flags

Issues

In some corner cases, including a drastic angle of walking trajectory, this method does not clear the correct flags

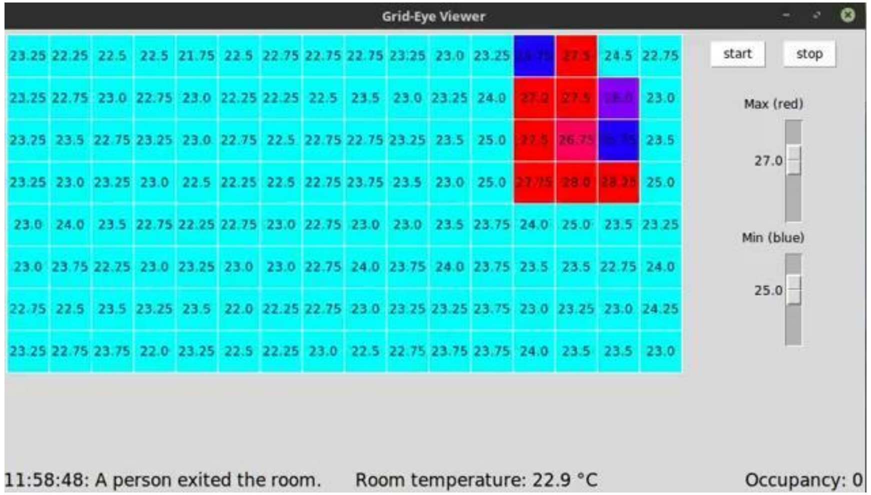

An example with a different column being fully flagged first


Incorrect flags cleared!

Clearing flags

Results

Before clearing out clusters



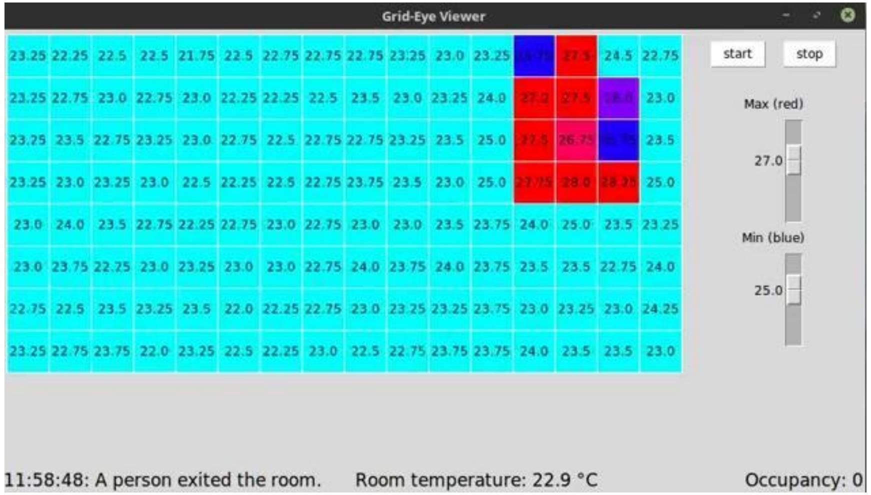
23.25	22.25	22.5	22.5	21.75	22.5	22.75	22.75	22.75	23.25	23.0	23.25	[illegible]	27.5	24.5	22.75
23.25	22.75	23.0	22.75	23.0	22.25	22.25	22.5	23.5	23.0	23.25	24.0	27.2	27.5	28.0	23.0
23.25	23.5	22.75	23.25	23.0	22.75	22.5	22.75	22.75	23.25	23.5	25.0	27.5	26.75	[illegible]	23.5
23.25	23.0	23.25	23.0	22.5	22.25	22.5	22.75	23.75	23.5	23.0	25.0	27.75	28.0	28.25	25.0
23.0	24.0	23.5	22.75	22.25	22.75	23.0	22.75	23.0	23.0	23.5	23.75	24.0	25.0	23.5	23.25
23.0	23.75	22.25	23.0	23.25	23.0	23.0	22.75	24.0	23.75	24.0	23.75	23.5	23.5	22.75	24.0
22.75	22.5	23.5	23.25	23.5	22.0	22.25	22.75	23.0	23.25	23.25	23.75	23.0	23.25	23.0	24.25
23.25	22.75	23.75	22.0	23.25	22.5	22.25	23.0	22.5	22.75	23.75	23.75	24.0	23.5	23.5	23.0

Max (red) 27.0

Min (blue) 25.0

11:58:48: A person exited the room. Room temperature: 22.9 °C Occupancy: 0

After clearing out clusters



23.25	22.25	22.5	22.5	21.75	22.5	22.75	22.75	22.75	23.25	23.0	23.25	[illegible]	27.5	24.5	22.75
23.25	22.75	23.0	22.75	23.0	22.25	22.25	22.5	23.5	23.0	23.25	24.0	27.2	27.5	28.0	23.0
23.25	23.5	22.75	23.25	23.0	22.75	22.5	22.75	22.75	23.25	23.5	25.0	27.5	26.75	[illegible]	23.5
23.25	23.0	23.25	23.0	22.5	22.25	22.5	22.75	23.75	23.5	23.0	25.0	27.75	28.0	28.25	25.0
23.0	24.0	23.5	22.75	22.25	22.75	23.0	22.75	23.0	23.0	23.5	23.75	24.0	25.0	23.5	23.25
23.0	23.75	22.25	23.0	23.25	23.0	23.0	22.75	24.0	23.75	24.0	23.75	23.5	23.5	22.75	24.0
22.75	22.5	23.5	23.25	23.5	22.0	22.25	22.75	23.0	23.25	23.25	23.75	23.0	23.25	23.0	24.25
23.25	22.75	23.75	22.0	23.25	22.5	22.25	23.0	22.5	22.75	23.75	23.75	24.0	23.5	23.5	23.0

Max (red) 27.0

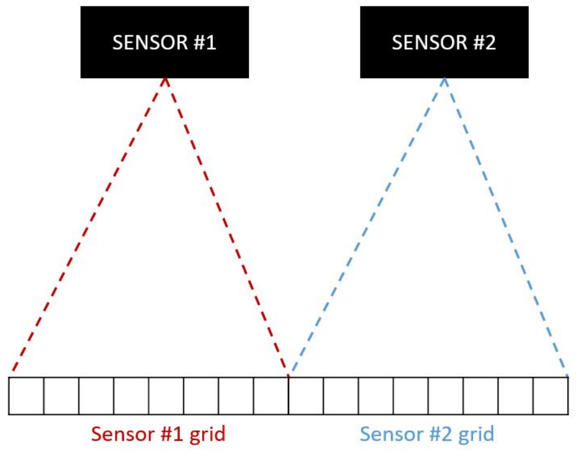
Min (blue) 25.0

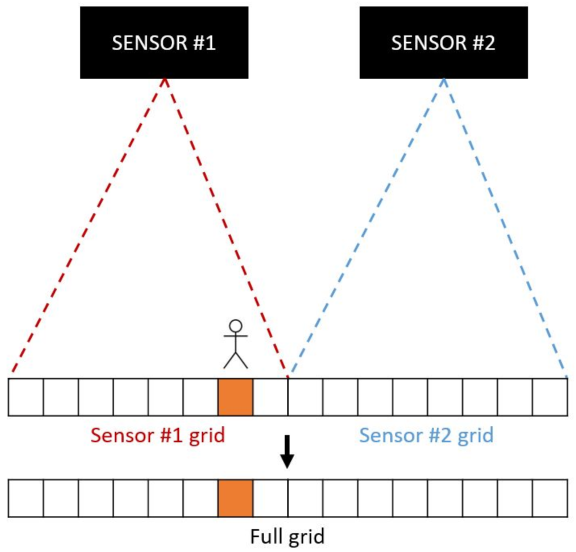
11:58:48: A person exited the room. Room temperature: 22.9 °C Occupancy: 0

Overlapping readings

Ideal case

Ideally, we could space each sensor out perfectly so that one sensor's view ends exactly where the next sensor's view begins:

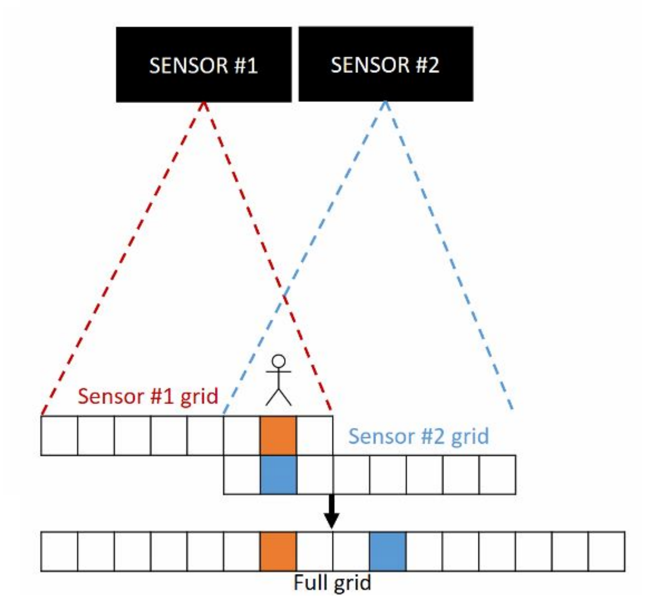




Overlapping readings

Real-life case

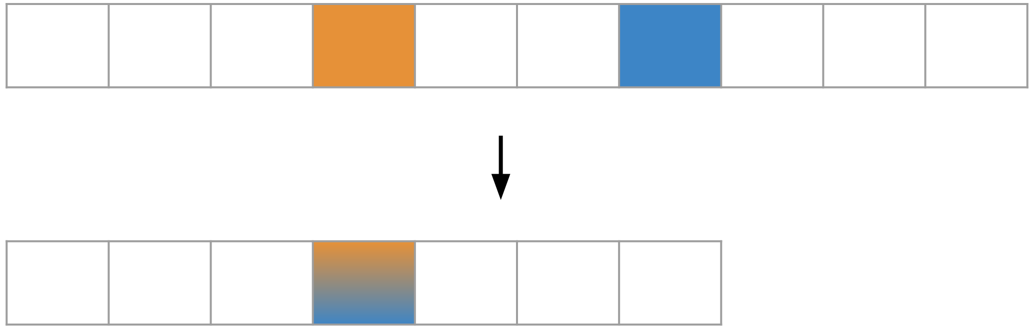
In reality, this can be difficult or impossible for many reasons (limited space, placement locations, slight movement over time, etc.):



Overlapping readings

Calibration

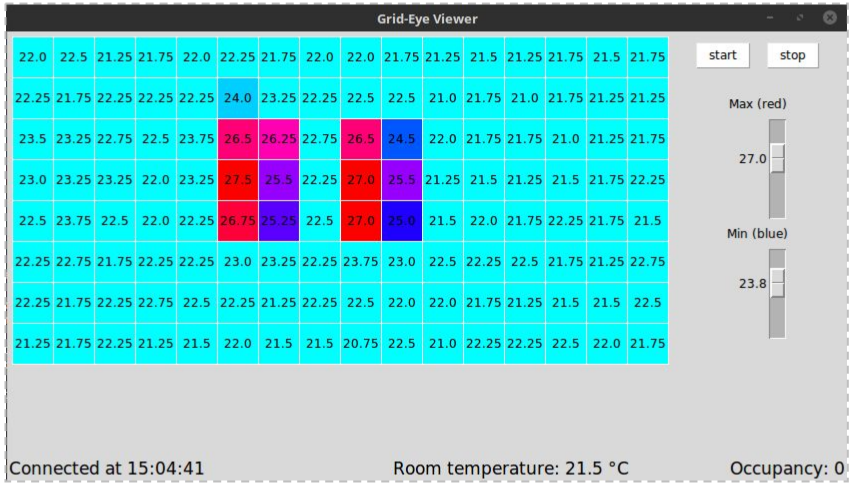
- If we have columns which overlap, we can eliminate the mirror effect by combining the columns.
- Replace them with the average of their two readings:



Overlapping readings

Results

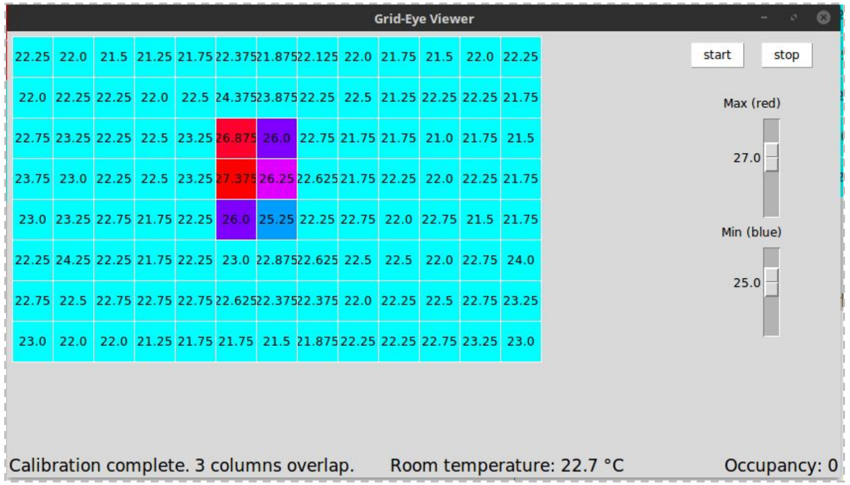
Before calibrating to eliminate overlaps



22.0	22.5	21.25	21.75	22.0	22.25	21.75	22.0	22.0	21.75	21.25	21.5	21.25	21.75	21.5	21.75
22.25	21.75	22.25	22.25	22.25	24.0	23.25	22.25	22.5	22.5	21.0	21.75	21.0	21.75	21.25	21.25
23.5	23.25	22.75	22.5	23.75	26.5	26.25	22.75	26.5	24.5	22.0	21.75	21.75	21.0	21.25	21.75
23.0	23.25	23.25	22.0	23.25	27.5	25.5	22.25	27.0	25.5	21.25	21.5	21.25	21.5	21.75	22.25
22.5	23.75	22.5	22.0	22.25	26.75	25.25	22.5	27.0	25.0	21.5	22.0	21.75	22.25	21.75	21.5
22.25	22.75	21.75	22.25	22.25	23.0	23.25	22.25	23.75	23.0	22.5	22.25	22.5	21.75	21.25	22.75
22.25	21.75	22.25	22.75	22.5	22.25	21.25	22.25	22.5	22.0	22.0	21.75	21.25	21.5	21.5	22.5
21.25	21.75	22.25	21.25	21.5	22.0	21.5	21.5	20.75	22.5	21.0	22.25	22.25	22.5	22.0	21.75

Connected at 15:04:41 Room temperature: 21.5 °C Occupancy: 0

After calibrating to eliminate overlaps



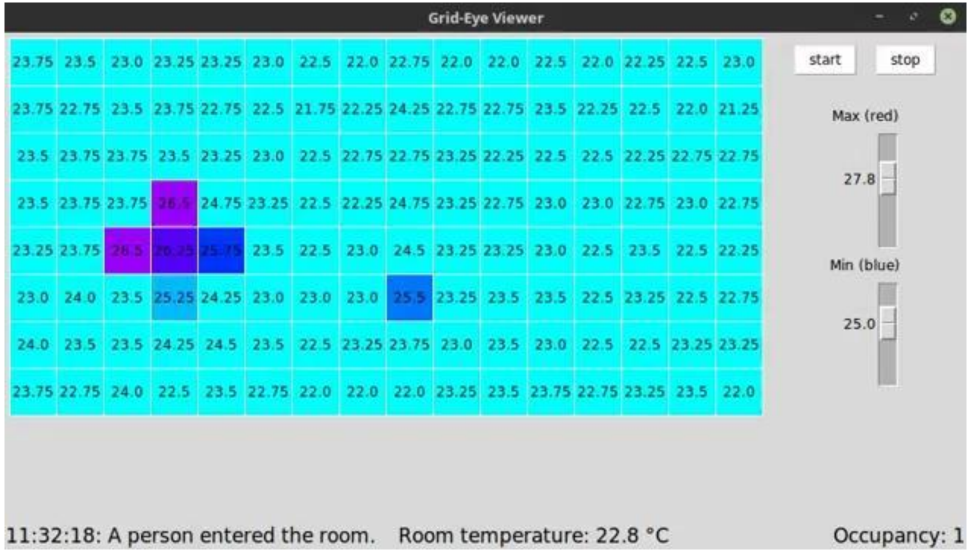
22.25	22.0	21.5	21.25	21.75	22.375	21.875	22.125	22.0	21.75	21.5	22.0	22.25
22.0	22.25	22.25	22.0	22.5	24.375	23.875	22.25	22.5	21.25	22.25	22.25	21.75
22.75	23.25	22.25	22.5	23.25	26.875	26.0	22.75	21.75	21.75	21.0	21.75	21.5
23.75	23.0	22.25	22.5	23.25	27.375	26.25	22.625	21.75	22.25	22.0	22.25	21.75
23.0	23.25	22.75	21.75	22.25	26.0	25.25	22.25	22.75	22.0	22.75	21.5	21.75
22.25	24.25	22.25	21.75	22.25	23.0	22.875	22.625	22.5	22.5	22.0	22.75	24.0
22.75	22.5	22.75	22.75	22.75	22.625	22.375	22.375	22.0	22.25	22.5	22.75	23.25
23.0	22.0	22.0	21.25	21.75	21.75	21.5	21.875	22.25	22.25	22.75	23.25	23.0

Calibration complete. 3 columns overlap. Room temperature: 22.7 °C Occupancy: 0

Overlapping readings

Results

Due to real-world conditions and a lack of symmetry, this solution is not perfect. The same object may be detectable in two columns, but the sensors will not read the same amount of heat from it, so the average isn't always best.



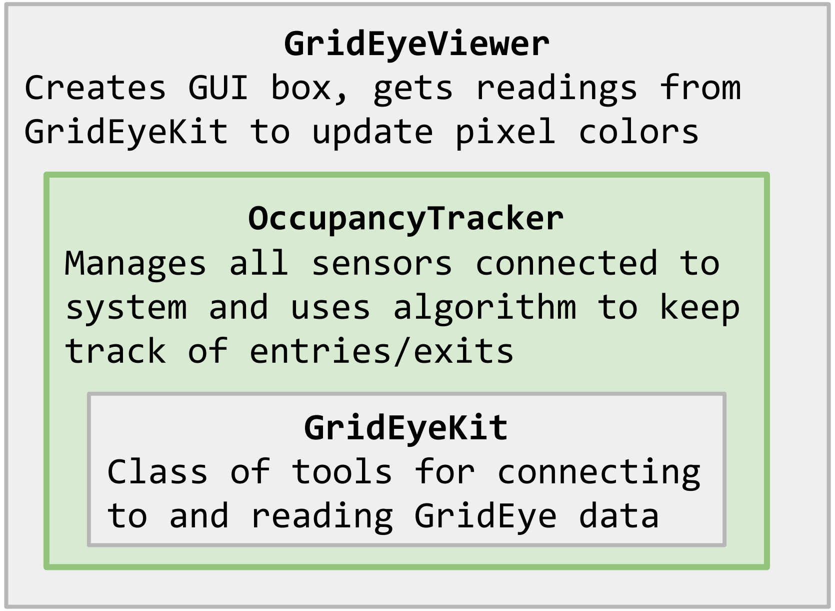
23.75	23.5	23.0	23.25	23.25	23.0	22.5	22.0	22.75	22.0	22.0	22.5	22.0	22.25	22.5	23.0
23.75	22.75	23.5	23.75	22.75	22.5	21.75	22.25	24.25	22.75	22.75	23.5	22.25	22.5	22.0	21.25
23.5	23.75	23.75	23.5	23.25	23.0	22.5	22.75	22.75	23.25	22.25	22.5	22.5	22.25	22.75	22.75
23.5	23.75	23.75	26.5	24.75	23.25	22.5	22.25	24.75	23.25	22.75	23.0	23.0	22.75	23.0	22.75
23.25	23.75	26.5	26.25	25.75	23.5	22.5	23.0	24.5	23.25	23.25	23.0	22.5	23.5	22.5	22.25
23.0	24.0	23.5	25.25	24.25	23.0	23.0	23.0	25.5	23.25	23.5	23.5	22.5	23.25	22.5	22.75
24.0	23.5	23.5	24.25	24.5	23.5	22.5	23.25	23.75	23.0	23.5	23.0	22.5	22.5	23.25	23.25
23.75	22.75	24.0	22.5	23.5	22.75	22.0	22.0	22.0	23.25	23.5	23.75	22.75	23.25	23.5	22.0

11:32:18: A person entered the room. Room temperature: 22.8 °C Occupancy: 1

Software overview

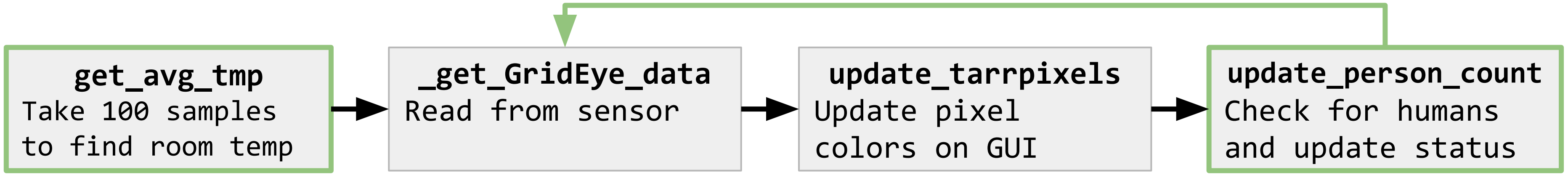
Software model | Class hierarchy

Class hierarchy, showing addition made to Grid-EYE reading and GUI software created by Alexander Hoch, which can be downloaded here.


GridEyeViewer
Creates GUI box, gets readings from GridEyeKit to update pixel colors
OccupancyTracker
Manages all sensors connected to system and uses algorithm to keep track of entries/exits
GridEyeKit
Class of tools for connecting to and reading GridEye data

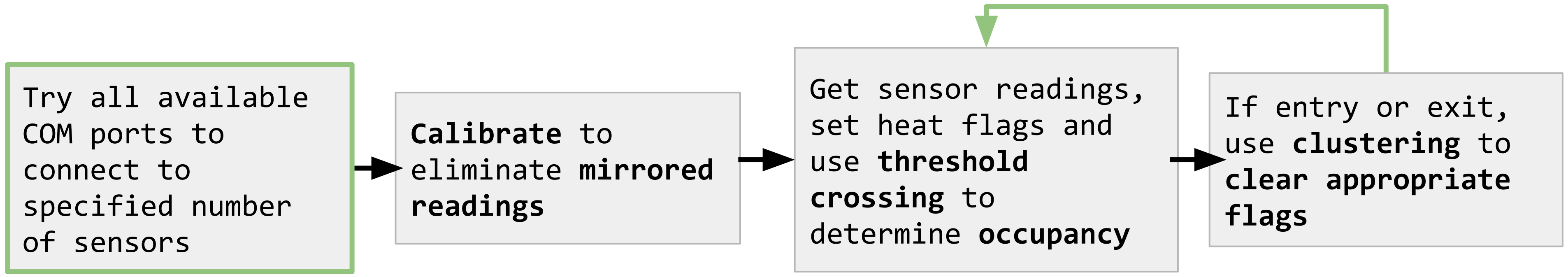
Software model | High-level flow

High-level functionality map, showing addition made to Grid-EYE reading and GUI software created by Alexander Hoch, which can be downloaded here.


get_avg_tmp
Take 100 samples to find room temp
_get_GridEye_data
Read from sensor
update_tarrpixels
Update pixel colors on GUI
update_person_count
Check for humans and update status

Software model | High-level flow

High-level functionality map, showing addition made to Grid-EYE reading and GUI software created by Alexander Hoch, which can be downloaded here.



Try all available COM ports to connect to specified number of sensors

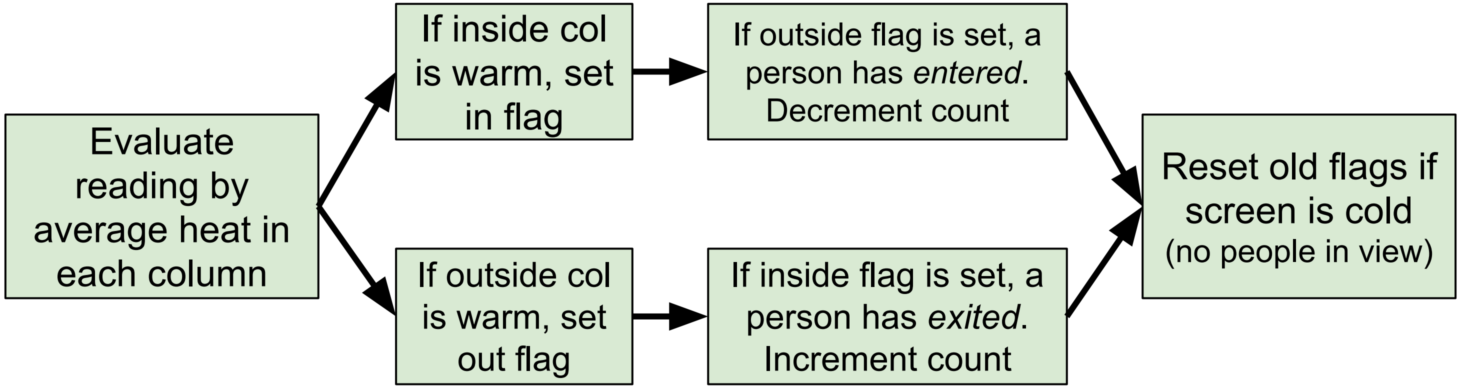
Calibrate to eliminate **mirrored readings**

Get sensor readings, set heat flags and use **threshold crossing** to determine **occupancy**

If entry or exit, use **clustering** to **clear appropriate flags**

Software model | update_person_count

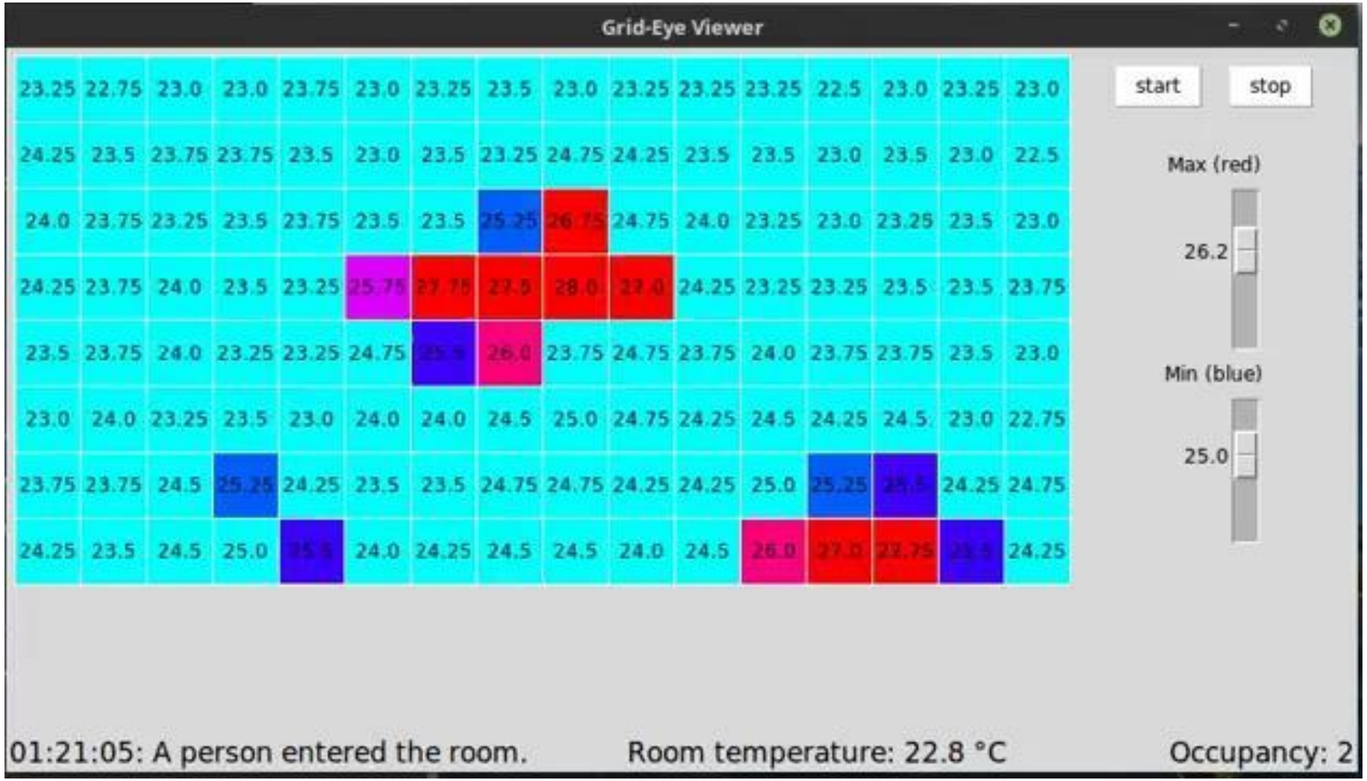
Like the thru-beam sensor approach, the program uses the fact that 2 thresholds will be crossed when a person passes through a doorway:


Evaluate reading by average heat in each column
If inside col is warm, set in flag
If outside col is warm, set out flag
If outside flag is set, a person has *entered*. Decrement count
If inside flag is set, a person has *exited*. Increment count
Reset old flags if screen is cold (no people in view)

This functionality was implemented with a new OccupancyTracker class which was made an object of the EvalKit class from the original code

Results

Final demonstration



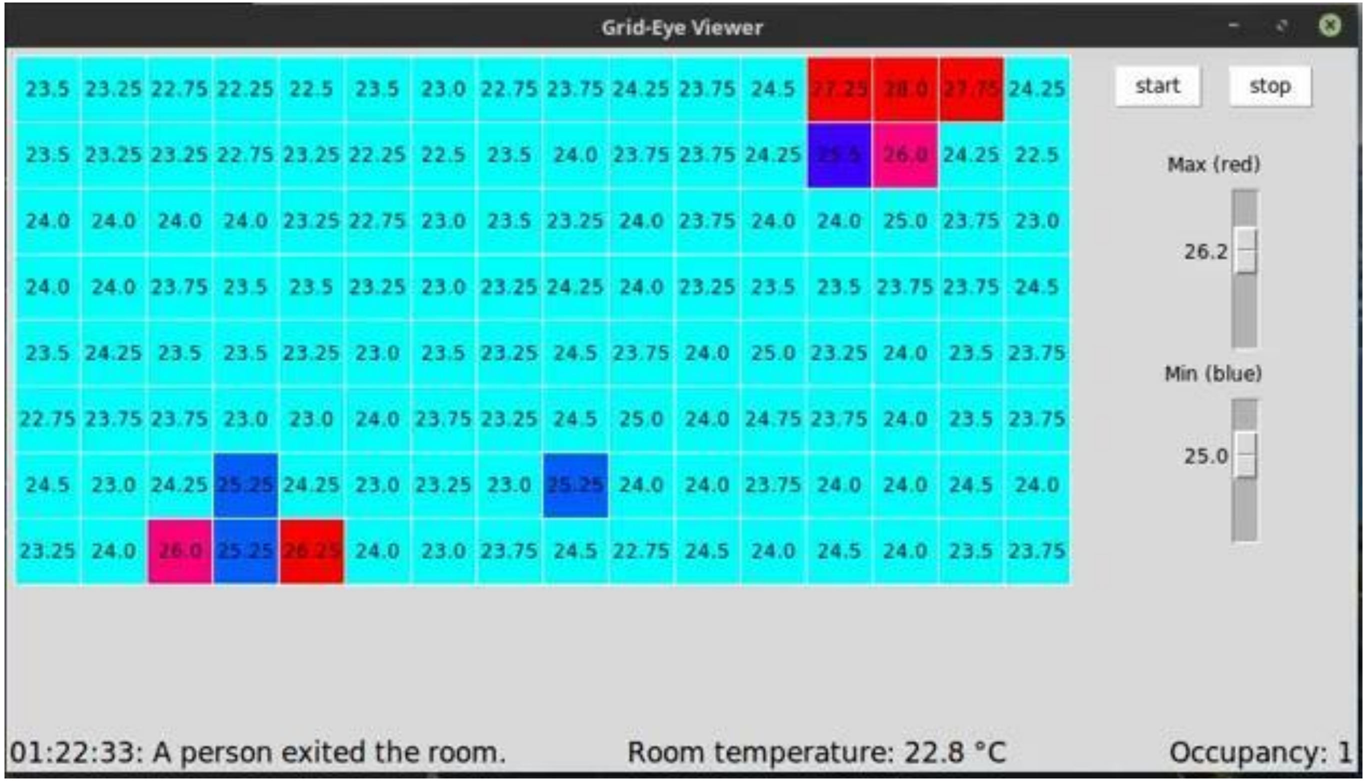
23.25	22.75	23.0	23.0	23.75	23.0	23.25	23.5	23.0	23.25	23.25	23.25	22.5	23.0	23.25	23.0
24.25	23.5	23.75	23.75	23.5	23.0	23.5	23.25	24.75	24.25	23.5	23.5	23.0	23.5	23.0	22.5
24.0	23.75	23.25	23.5	23.75	23.5	23.5	25.25	26.75	24.75	24.0	23.25	23.0	23.25	23.5	23.0
24.25	23.75	24.0	23.5	23.25	25.75	27.75	27.5	28.0	27.0	24.25	23.25	23.25	23.5	23.5	23.75
23.5	23.75	24.0	23.25	23.25	24.75	25.5	26.0	23.75	24.75	23.75	24.0	23.75	23.75	23.5	23.0
23.0	24.0	23.25	23.5	23.0	24.0	24.0	24.5	25.0	24.75	24.25	24.5	24.25	24.5	23.0	22.75
23.75	23.75	24.5	25.25	24.25	23.5	23.5	24.75	24.75	24.25	24.25	25.0	25.25	25.5	24.25	24.75
24.25	23.5	24.5	25.0	25.5	24.0	24.25	24.5	24.5	24.0	24.5	26.0	27.0	27.75	25.5	24.25

Max (red) 26.2

Min (blue) 25.0

01:21:05: A person entered the room. Room temperature: 22.8 °C Occupancy: 2

Final demonstration



23.5	23.25	22.75	22.25	22.5	23.5	23.0	22.75	23.75	24.25	23.75	24.5	27.25	28.0	27.75	24.25
23.5	23.25	23.25	22.75	23.25	22.25	22.5	23.5	24.0	23.75	23.75	24.25	25.5	26.0	24.25	22.5
24.0	24.0	24.0	24.0	23.25	22.75	23.0	23.5	23.25	24.0	23.75	24.0	24.0	25.0	23.75	23.0
24.0	24.0	23.75	23.5	23.5	23.25	23.0	23.25	24.25	24.0	23.25	23.5	23.5	23.75	23.75	24.5
23.5	24.25	23.5	23.5	23.25	23.0	23.5	23.25	24.5	23.75	24.0	25.0	23.25	24.0	23.5	23.75
22.75	23.75	23.75	23.0	23.0	24.0	23.75	23.25	24.5	25.0	24.0	24.75	23.75	24.0	23.5	23.75
24.5	23.0	24.25	25.25	24.25	23.0	23.25	23.0	25.25	24.0	24.0	23.75	24.0	24.0	24.5	24.0
23.25	24.0	26.0	25.25	26.25	24.0	23.0	23.75	24.5	22.75	24.5	24.0	24.5	24.0	23.5	23.75

01:22:33: A person exited the room. Room temperature: 22.8 °C Occupancy: 1

Key contributions, Spring 2021

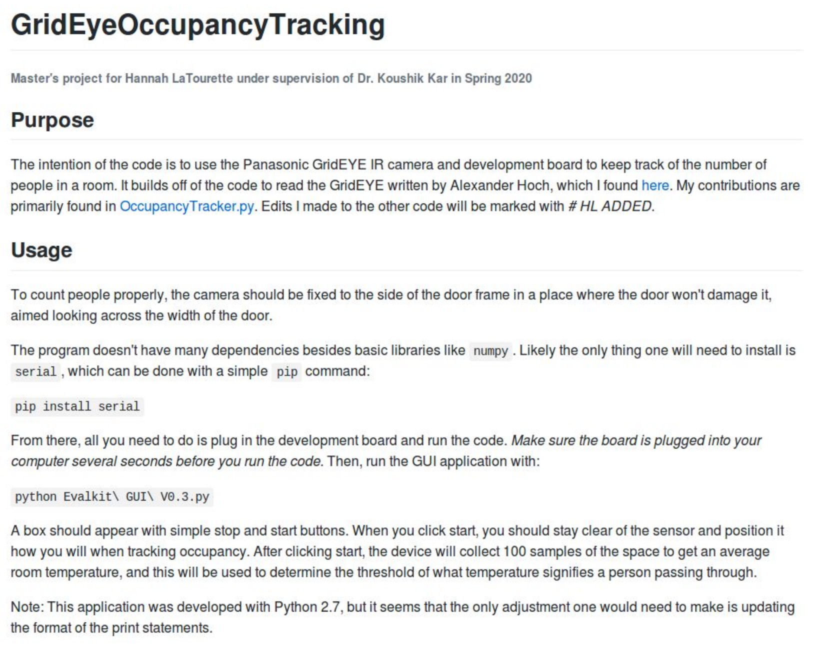
- Integration of multiple sensors and corresponding GUI display
- Dashboard which displays latest event, room temperature, and current occupancy to user at all time
- Two-dimensional heat flag array which allows for detection of the entry and exit of multiple people at once
- Effective handling for known overlapping columns

- Cluster detection which ensures appropriate flags are cleared when a person has passed the threshold (works outside of certain corner cases)

- Automatic detection of overlapping columns (somewhat work in progress)

Documentation

Project available on Github at https://github.com/hannahlatourette/GridEyeOccupancyTracking



GridEyeOccupancyTracking

Master's project for Hannah LaTourette under supervision of Dr. Koushik Kar in Spring 2020

Purpose

The intention of the code is to use the Panasonic GridEYE IR camera and development board to keep track of the number of people in a room. It builds off of the code to read the GridEYE written by Alexander Hoch, which I found here. My contributions are primarily found in OccupancyTracker.py. Edits I made to the other code will be marked with *# HL ADDED*.

Usage

To count people properly, the camera should be fixed to the side of the door frame in a place where the door won't damage it, aimed looking across the width of the door.

The program doesn't have many dependencies besides basic libraries like `numpy`. Likely the only thing one will need to install is `serial`, which can be done with a simple `pip` command:

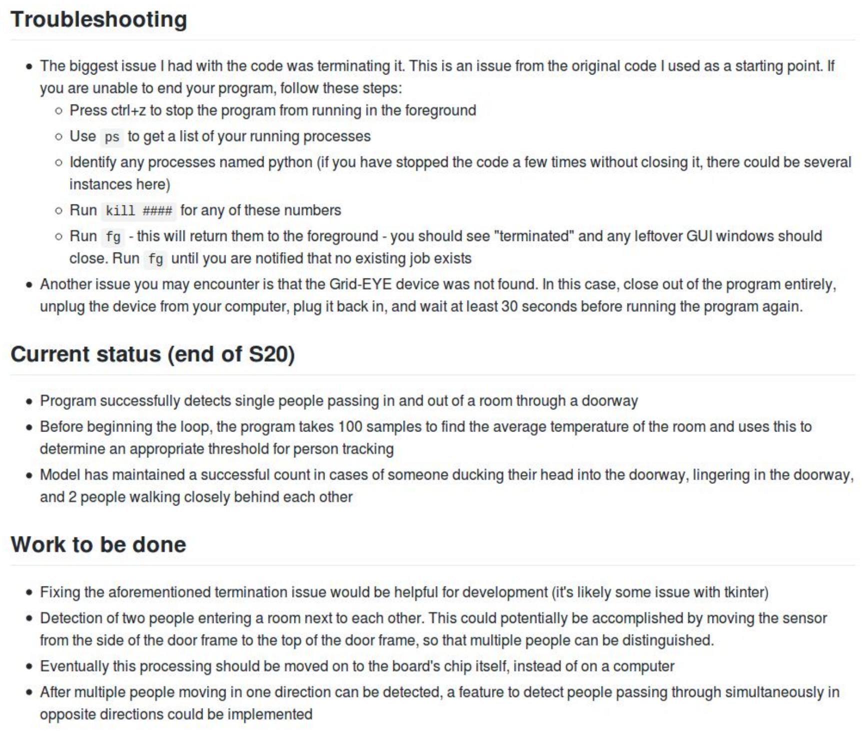
```
pip install serial
```

From there, all you need to do is plug in the development board and run the code. *Make sure the board is plugged into your computer several seconds before you run the code.* Then, run the GUI application with:

```
python Evalkit\ GUI\ V0.3.py
```

A box should appear with simple stop and start buttons. When you click start, you should stay clear of the sensor and position it how you will when tracking occupancy. After clicking start, the device will collect 100 samples of the space to get an average room temperature, and this will be used to determine the threshold of what temperature signifies a person passing through.

Note: This application was developed with Python 2.7, but it seems that the only adjustment one would need to make is updating the format of the print statements.



Troubleshooting

- The biggest issue I had with the code was terminating it. This is an issue from the original code I used as a starting point. If you are unable to end your program, follow these steps:
  - Press ctrl+z to stop the program from running in the foreground
  - Use `ps` to get a list of your running processes
  - Identify any processes named python (if you have stopped the code a few times without closing it, there could be several instances here)
  - Run `kill ####` for any of these numbers
  - Run `fg` - this will return them to the foreground - you should see "terminated" and any leftover GUI windows should close. Run `fg` until you are notified that no existing job exists
- Another issue you may encounter is that the Grid-EYE device was not found. In this case, close out of the program entirely, unplug the device from your computer, plug it back in, and wait at least 30 seconds before running the program again.

Current status (end of S20)

- Program successfully detects single people passing in and out of a room through a doorway
- Before beginning the loop, the program takes 100 samples to find the average temperature of the room and uses this to determine an appropriate threshold for person tracking
- Model has maintained a successful count in cases of someone ducking their head into the doorway, lingering in the doorway, and 2 people walking closely behind each other

Work to be done

- Fixing the aforementioned termination issue would be helpful for development (it's likely some issue with tkinter)
- Detection of two people entering a room next to each other. This could potentially be accomplished by moving the sensor from the side of the door frame to the top of the door frame, so that multiple people can be distinguished.
- Eventually this processing should be moved on to the board's chip itself, instead of on a computer
- After multiple people moving in one direction can be detected, a feature to detect people passing through simultaneously in opposite directions could be implemented

Next steps

- More proficient overlapping detection
- On-chip analysis & Bluetooth integration
- Student will work over the summer on GUI updates (manual occupancy reset, push notifications, etc.)
- Bluetooth communication in sensors may be incorporated into future labs for ECSE 4660/6660 Internetworking of Things
- When students can return to campus labs, a solution using thru-beam sensors may also be investigated and implemented

References

- Corridor & Hallway Dimension | Dimension Guide
- The Important Factors of Corridors Settings in Shopping Center Design | A. Kusumowidagdo
- Reducing Battery Consumption | Temboo

Links

Github page with source code and documentation

FLOWCODE

PRIVACY.FLOWCODE.COM

These presentation slides

FLOWCODE

PRIVACY.FLOWCODE.COM

Thank you!

Questions?