College of Engineering, Design and Computing
UNIVERSITY OF COLORADO DENVER

PEOPLE/TRACKING LOCATION & ANALYSIS

ELEC 4319 – Senior Design Project II

Team leader: Patrick Bales-Parks
+1 (720) 425-4916
patrick.bales-parks@ucdenver.edu

Fadhel Alkhater
+1 (503) 999-9234
fadhel.alkhater@ucdenver.edu

Huseen Alramzi
+1 (720) 266-3466
huseen.alramzi@ucdenver.edu

Chance Prem
+1 (843) 259-4048
chance.prem@ucdenver.edu

Amanda Rowsell
+1 (720) 413-4509
amanda.rowsell@ucdenver.edu

Jonathan Smith
+1 (757) 693-0822
jonathan.smith@ucdenver.edu

University of Colorado Denver, Department of Electrical Engineering | 1200 Larimer Street Suite 3034 Denver, CO 80204

Description of Project:

In the spring of 2020, the COVID-19 virus began a worldwide pandemic that sent the world into lockdowns. The number of people in a room was limited and people when they went outside of their homes were required to stay a certain distance away from others. To have an individual person to enforce these rules or watch where everyone went was no feasible creating a problem that needed to be solved. The purpose of our project is to implement an inexpensive, effective, and non-invasive solution to track individuals in an area, for the purpose of increasing health and safety by reducing the risk of virus transmission through data collection and analysis.

This goal is to be achieved by implementing computer vision and machine learning techniques. These techniques are implemented using “edge”, in the field, devices equipped with cameras and the processing power necessary to run calculation intensive algorithms. Our multi-camera system will map locations of people in the building on a 2D floorplan to assist First Responders in emergency situations and provide foot traffic data to campus planners and other industry professionals. Our system allows the user to determine if they want real time location of each person or if they want to see the traffic patterns of people and where social distancing violations occurred to determine any areas of concern.

Design Methodology, Constraints, Specifications, and System requirements (CLO 1):

To design our system, we started by using a Mind Map. A Mind Map allowed us to analyze our problem statement and start to determine all the necessary requirements our system would need to satisfy the problem statement. From our Mind Map we determined that tracking, alerts, sensing, and security were the main requirements we would need to meet for a successful prototype.

Figure 1. Mind Map

For the next part of the Design Process, we used a Journey Map to walk through the process of how someone would come to need our product to help us understand any additional requirements that may be needed or how we should adjust others to have a successful prototype. We started by having a “user” determine that they needed a system that kept track of the number of people in a room or a person’s physical location in a room. We then showed how our system would just need to be installed and aligned for the room it was in and then they would be able to receive data about where a person in their room/business was and past foot traffic

patterns. We also showed how the system could be given to law enforcement, paramedics, or firefighters to help with response to an emergency.

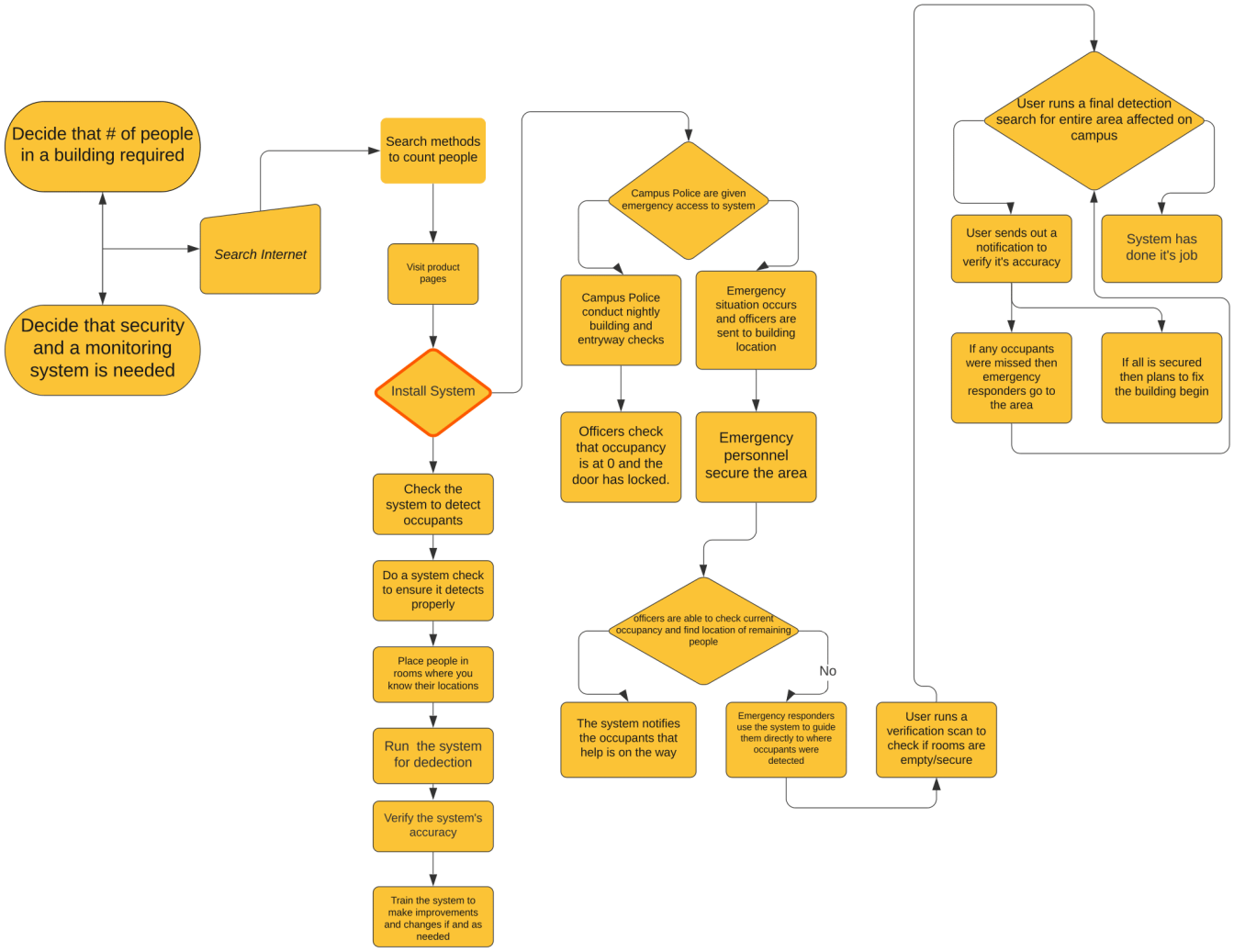


Figure 2. Journey Map

Engineering Documents (CLO 3):

The People/ Location and analysis team is following specific course steps to generate the final high-level design on their project. The steps are illustrated in the chart below (figure flow 2). The system hierarchy begins with the video stream input that is generated from the camera feed. The feed was initially calibrated using a make-shift checkerboard, and then depth maps could be generated from the video data. This data was then sent to the central Linux server computer. Once the data was received by the central computer, person detection and counting algorithms were applied and depth data was analyzed. The data analysis being performed on the central computer included 3D to 2D pixel and location mapping, object path prediction mapping, and display of detected person centroid bounding boxes in rewritten videos.

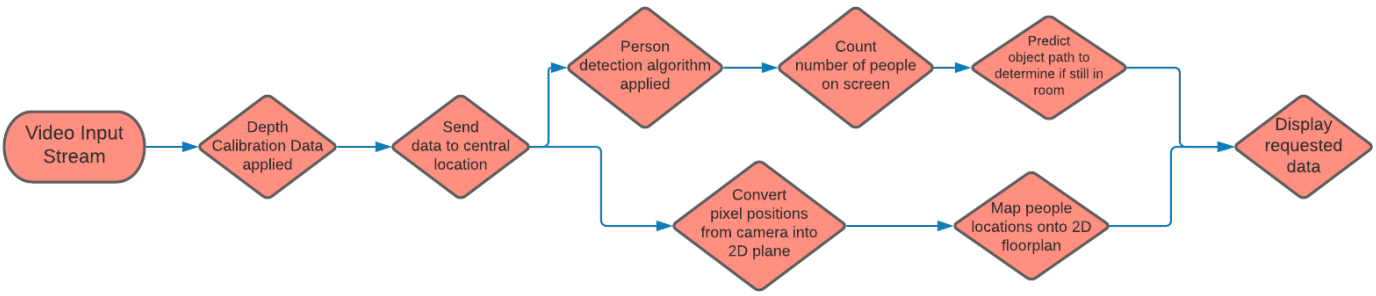
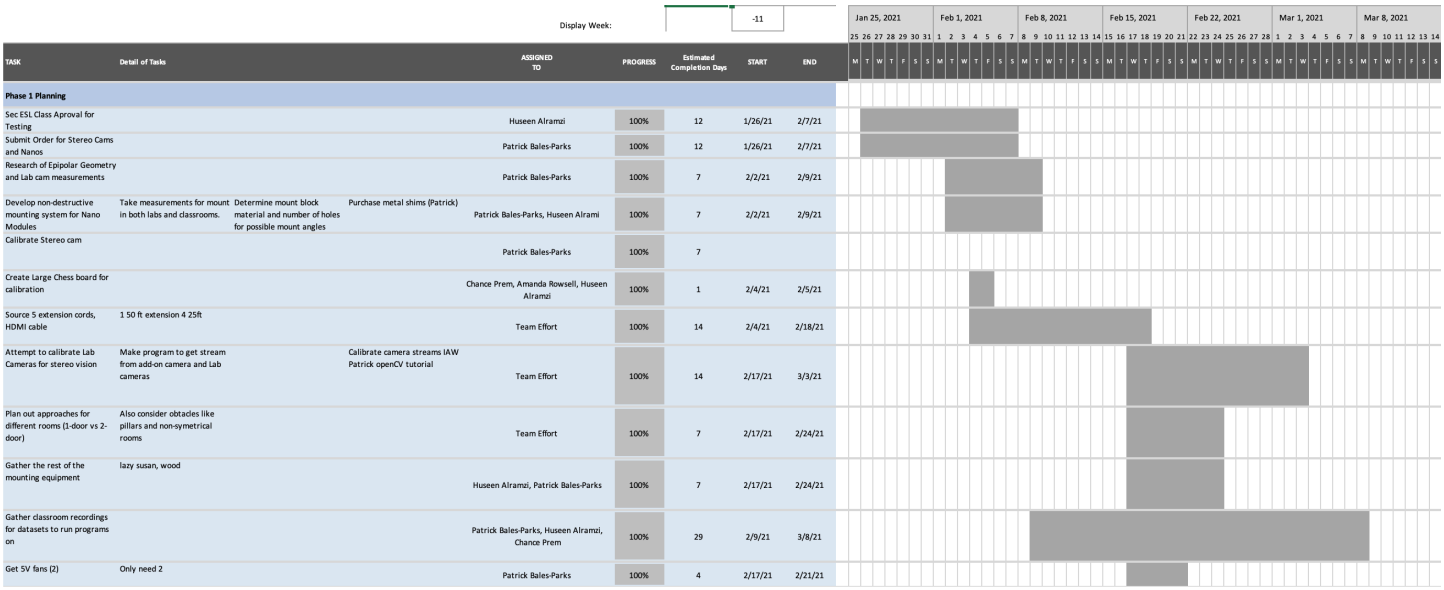


Figure 3. Flow chart

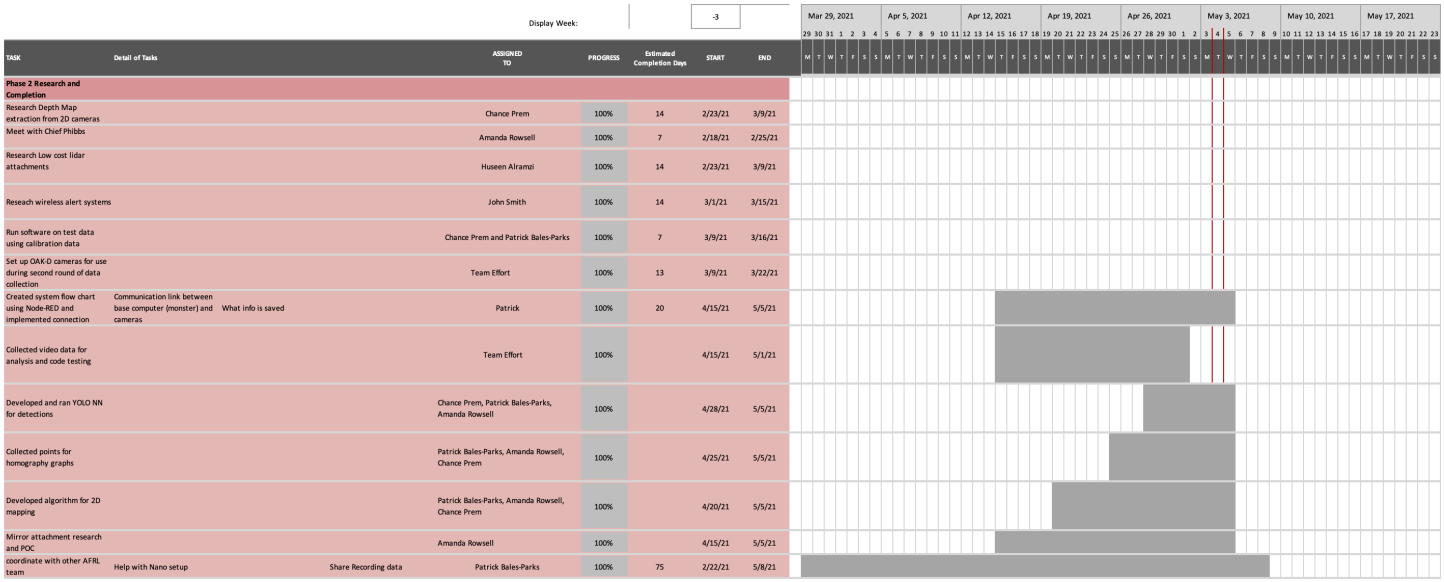
The Gantt charts were used to schedule out how our team would accomplish all the tasks required to implement our design. It tracked task start and end dates and allowed our team to assign tasks to individuals to ensure timely progress of our system.



Display Week: -11

TASK	Detail of Tasks			ASSIGNED TO	PROGRESS	Estimated Completion Days	START	END
Phase 1 Planning								
Sec ESL Class Aproval for Testing				Huseen Alramzi	100%	12	1/26/21	2/7/21
Submit Order for Stereo Cams and Nanos				Patrick Bales-Parks	100%	12	1/26/21	2/7/21
Research of Epipolar Geometry and Lab cam measurements				Patrick Bales-Parks	100%	7	2/2/21	2/9/21
Develop non-destructive mounting system for Nano Modules	Take measurements for mount in both labs and classrooms.	Determine mount block material and number of holes for possible mount angles	Purchase metal shims (Patrick)	Patrick Bales-Parks, Huseen Alrami	100%	7	2/2/21	2/9/21
Calibrate Stereo cam				Patrick Bales-Parks	100%	7		
Create Large Chess board for calibration				Chance Prem, Amanda Rowsell, Huseen Alramzi	100%	1	2/4/21	2/5/21
Source 5 extension cords, HDMI cable	1 50 ft extension 4 25ft			Team Effort	100%	14	2/4/21	2/18/21
Attempt to calibrate Lab Cameras for stereo vision	Make program to get stream from add-on camera and Lab cameras		Calibrate camera streams IAW Patrick openCV tutorial	Team Effort	100%	14	2/17/21	3/3/21
Plan out approaches for different rooms (1-door vs 2-door)	Also consider obtacles like pillars and non-symetrical rooms			Team Effort	100%	7	2/17/21	2/24/21
Gather the rest of the mounting equipment	lazy susan, wood			Huseen Alramzi, Patrick Bales-Parks	100%	7	2/17/21	2/24/21
Gather classroom recordings for datasets to run programs on				Patrick Bales-Parks, Huseen Alramzi, Chance Prem	100%	29	2/9/21	3/8/21
Get 5V fans (2)	Only need 2			Patrick Bales-Parks	100%	4	2/17/21	2/21/21

Figure 4. Gantt Chart Phase 1



Display Week: -3

TASK	Detail of Tasks			ASSIGNED TO	PROGRESS	Estimated Completion Days	START	END
Phase 2 Research and Completion								
Research Depth Map extraction from 2D cameras				Chance Prem	100%	14	2/23/21	3/9/21
Meet with Chief Phibbs				Amanda Rowsell	100%	7	2/18/21	2/25/21
Research Low cost lidar attachments				Huseen Alramzi	100%	14	2/23/21	3/9/21
Reseach wireless alert systems				John Smith	100%	14	3/1/21	3/15/21
Run software on test data using calibration data				Chance Prem and Patrick Bales-Parks	100%	7	3/9/21	3/16/21
Set up OAK-D cameras for use during second round of data collection				Team Effort	100%	13	3/9/21	3/22/21
Created system flow chart using Node-RED and implemented connection	Communication link between base computer (monster) and cameras	What info is saved		Patrick	100%	20	4/15/21	5/5/21
Collected video data for analysis and code testing				Team Effort	100%		4/15/21	5/1/21
Developed and ran YOLO NN for detections				Chance Prem, Patrick Bales-Parks, Amanda Rowsell	100%		4/28/21	5/5/21
Collected points for homography graphs				Patrick Bales-Parks, Amanda Rowsell, Chance Prem	100%		4/25/21	5/5/21
Developed algorithm for 2D mapping				Patrick Bales-Parks, Amanda Rowsell, Chance Prem	100%		4/20/21	5/5/21
Mirror attachment research and POC				Amanda Rowsell	100%		4/15/21	5/5/21
coordinate with other AFRL team	Help with Nano setup	Share Recording data		Patrick Bales-Parks	100%	75	2/22/21	5/8/21

Figure 5. Gantt Chart Phase 2

Fall 2020 Semester

Name	Quantity	Cost	total
JetsonNano Developer Kit V3	2	$99.00	-$198.00
64GB microSD card (class 10)	2	$29.95	-$59.90
Camera Module	2	$29.99	-$59.98
Jetson Wifi modules	5	$24.99	-$124.95
Jetson Case	5	$19.99	-$99.95
Budget	6	-$100.00	$600.00
Total			+ $57.22

Spring 2021 Semester

Name	Quantity	Cost	total
Jetson Nano Development Pack (Type D), with Binocular Camera, TF Card	2	$173.79	-$347.58
IMX219-83 Stereo Camera, 8MP Binocular Camera Module, Depth Vision	3	$43.49	-$130.47
Jetson Nano Metal Case (C), Camera Holder, Internal Fan Design	5	$12.99	-$64.95
Budget	6	-$100.00	$600.00
Total			+ $57.00

Figure 6. Budget Summary

We used the floor plan schematics to measure out the dimensions of the room and specific points that were identified to create the homograph transform needed to map the 3D camera locations over to the 2D floorplan.

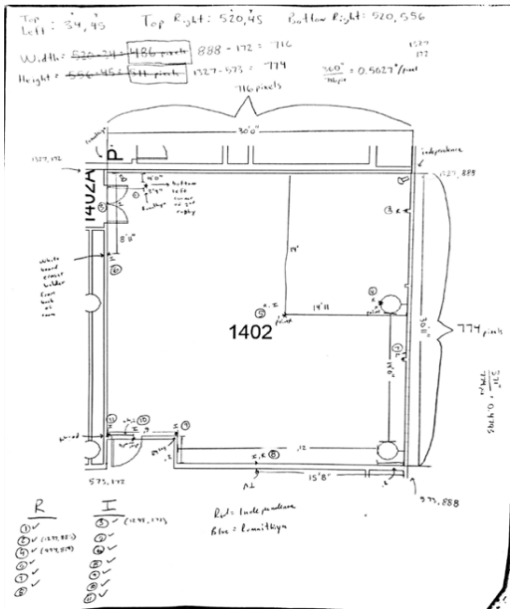


Figure 7. Rooms measurement

Node-RED Machine-to-Machine browser-based connection system and program flow implemented with MQTT (mosquito) communication protocols to allow for simultaneous command of 4 camera modules.

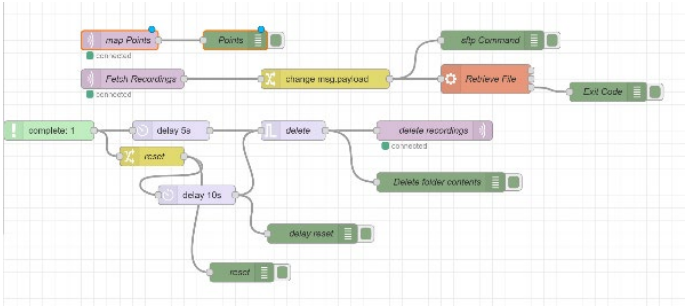


Figure 8a. Node Red

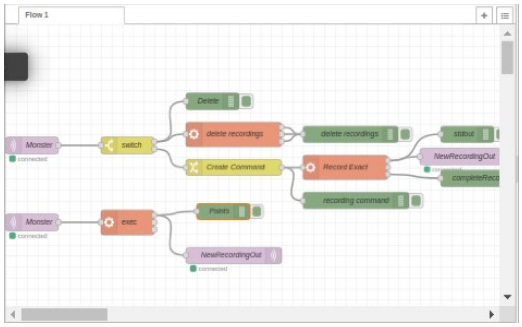


Figure 8b. Node Red

Below are the images of our python scripts used to achieve this project. Figure 9 shows the libraries that were used in the scripts. Figure 10 shows how we input the pixel points from the camera and the corresponding pixel points from the floorplan. These were then fed into a function that created a transformation matrix that would be applied to the centroid locations from our person detecting program. Figure 11 shows how the transform was implemented in the person detection script to map the points to the floorplan. Figure 12 then shows how MQTT protocol was used to simultaneously record a set size video from 4 different cameras.

```
# import the necessary packages
from pyimagesearch import social_distancing_config as config
from pyimagesearch.detection import detect_people
from scipy.spatial import distance as dist
import numpy as np
import argparse
import imutils
import cv2
import os

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", type=str, default="",
    help="path to (optional) input video file")
ap.add_argument("-i1", "--input1", type=str, default="",
    help="path to (optional) input1 video file")
ap.add_argument("-i2", "--input2", type=str, default="",
    help="path to (optional) input2 video file")
ap.add_argument("-i3", "--input3", type=str, default="",
    help="path to (optional) input3 video file")
ap.add_argument("-o", "--output", type=str, default="",
    help="path to (optional) output video file")
ap.add_argument("-o1", "--output1", type=str, default="",
    help="path to (optional) output1 video file")
ap.add_argument("-d", "--display", type=int, default=1,
    help="whether or not output frame should be displayed")
args = vars(ap.parse_args())
```

Figure 9. Library files

```
67
68  bob_pts_src = np.float32([[52,1035],[52,340],[937,237],[1512,278],[1515,667],[1849,667],[1849,1035]])
69  bob_pts_dst = np.float32([[601,322],[427,590],[21,530],[21,234],[457,139],[457,97],[462,105]])
70  bry_pts_src = np.float32([[3,1078],[307,127],[1903,127],[1903,1078]])
71  bry_pts_dst = np.float32([[21,436],[21,90],[613,43],[206,475]])
72  nn_pts_src = np.float32([[9,1068],[9,250],[69,87],[451,88],[1504,160],[1689,376],[1689,1068]])
73  nn_pts_dst = np.float32([[454,494],[187,336],[21,219],[76,90],[508,105],[583,203],[613,415]])
74  ind_pts_src = np.float32([[558,1024],[558,184],[1273,133],[1476, 175],[1476, 1029]])
75  ind_pts_dst = np.float32([[298, 111],[690, 245],[677, 590],[539, 590],[223, 244]])
76
77  bryh,_ = cv2.findHomography(bry_pts_src,bry_pts_dst,method=0)
78  bobh,_ = cv2.findHomography(bob_pts_src, bob_pts_dst, method = 0)
79  indh,_ = cv2.findHomography(ind_pts_src, ind_pts_dst, method = 0)
80  nnh,_ = cv2.findHomography(nn_pts_src, nn_pts_dst, method = 0)
```

Figure 10. Homography creation

```
230     if len(results2) > 0:
231         for (y, (prob2, bbox2, centroid2)) in enumerate(results2):
232             # extract the bounding box and centroid coordinates, then
233             # initialize the color of the annotation
234             (startX2, startY2, endX2, endY2) = bbox2
235             (cX2, cY2) = centroid2
236             cx2 = centroid2[0]
237             cy2 = centroid2[1]
238             color = (0, 255, 0)
239             ind_pt_test = np.array([[cx2-15,cy2]], dtype='float32')
240             ind_pt_test = np.array([ind_pt_test])
241
242             ind_out_pt = cv2.perspectiveTransform(ind_pt_test, indh)
243
244             if count % 5 == 0:
245                 image = original_image.copy()
246             image = cv2.circle(image, (ind_out_pt[0][0][0], ind_out_pt[0][0][1]), radius=4, color=(0, 128, 255), thickness=4)
247             # if the index pair exists within the violation set, then
248             # update the color
249             if y in violate:
250                 color = (0, 0, 255)
251                 image = cv2.circle(image, (ind_out_pt[0][0][0], ind_out_pt[0][0][1]), radius=2, color=(0, 0, 255), thickness=2)
252                 # draw (1) a bounding box around the person and (2) the
253                 # centroid coordinates of the person,
254             cv2.rectangle(frame2, (startX2, startY2), (endX2, endY2), color, 2)
255             cv2.circle(frame2, (cX2, cY2), 5, color, 1)
256             # draw the total number of social distancing violations on the
257             # output frame
```

Figure 11. Homography mapping implementation

```
plta@u109490:~$ mosquitto_pub -h localhost -t "nano2609" -m "record VideoName for 60"
```

Figure 12. MQTT protocol

Computer Design Tools (CLO 4):

Mounting systems for NC 1402 and NC 2609 camera modules were created using AutoCAD software. They needed to be able to hold the weight of the camera module and maintain a steady base to ensure that the camera position did not change over the course of recordings. We also needed to have a swivel in the mount to adjust the field of view of the cameras.

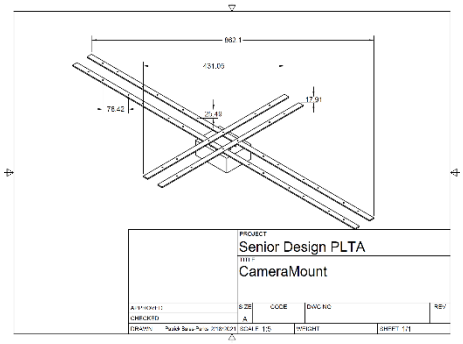


Figure 13. Camera mount schematic

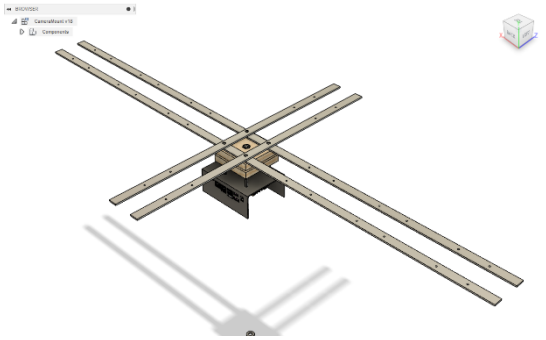
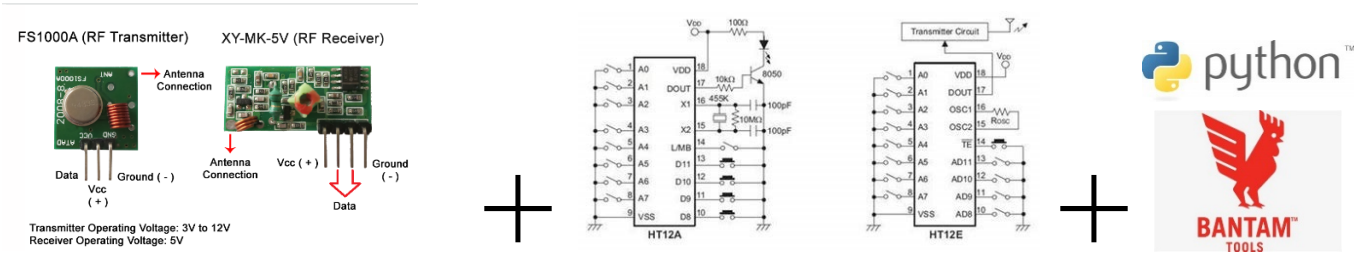
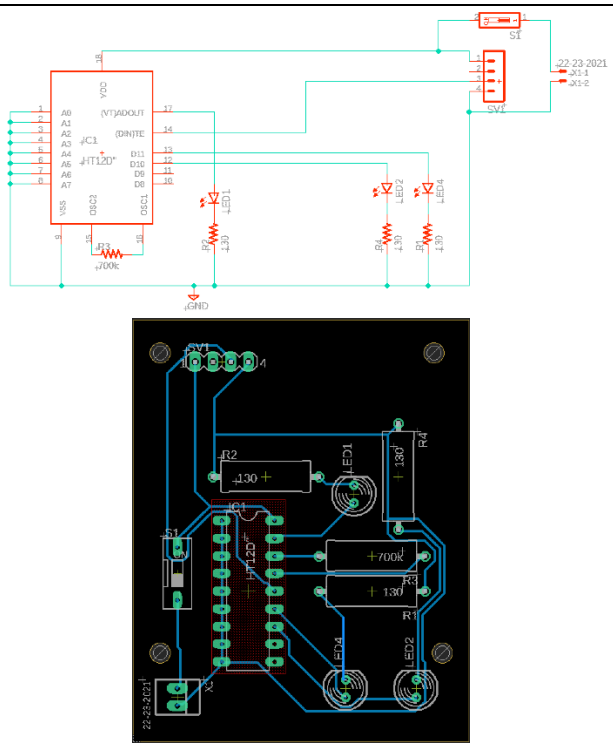
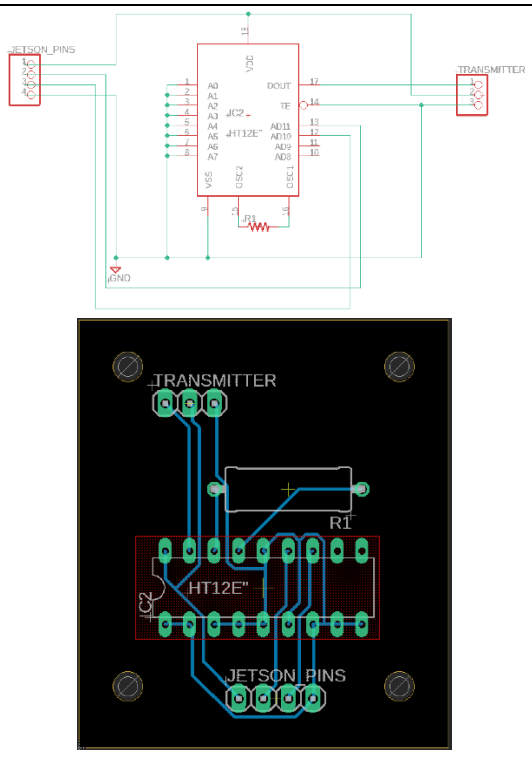


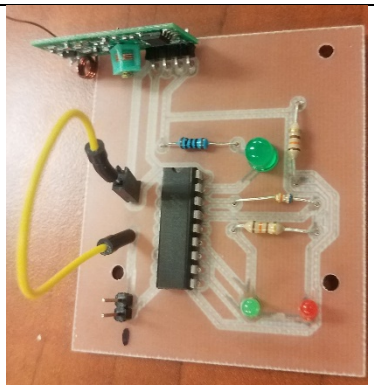
Figure 14. AutoCAD camera mount

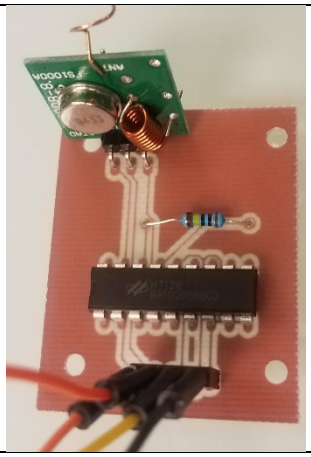
The wireless alert system below was designed to interface with the Jetson Nano GPIO pins within the camera modules, allowing the communication of up to 4 bits of data when a change in room status is detected. For example, if the Nano determines that a pre-determined occupancy limit has been reached, it will transmit a 'low' signal for the green LED and a 'high' signal for the red LED, therefore illuminating only the red LED and allowing people outside of the room to know it is currently unsafe to enter. The transmitter design utilizes a 433 MHz transmitter module connected to a Holtek HT12E encoder IC which allows for up to 8 address bits to be selected and transmits the status through its quarter wave antenna on 4 data pins. The receiver circuit consists of a 433 MHz receiver module with antenna, HT12D decoder IC, indicator LEDs, a power switch, a 6 V battery

pack, and corresponding resistors. The receiver circuit will illuminate the green status LED by default until receiving a signal from the transmitter directing it to switch. The circuits were successfully prototyped on breadboards before being translated to an Eagle software schematic and cut into PCBs. The main issue when prototyping was unreliable receiver modules. Out of 9 total modules, 3 were able to reliably receive at a reasonable distance. Ultimately, this design allows us to wirelessly communicate 2 data bits which directly control 2 LEDs over approximately 5 meters, allowing for the Jetson Nano to indicate room status and therefore decreases the risk of virus transmission.



Receiver PCB	Transmitter PCB
	





Throughout the interactions the team maintained with the Air Force Research Laboratory, Campus Planning, and Auraria Security sponsors, we came to understand that upgrading an entire camera system was not only expensive but also cumbersome. In order to account for this, the team investigated ways of reconfiguring existing camera systems in order to utilize the benefits of stereo camera systems. The benefits of a stereo camera system the team particularly focused on were depth data as well as increasing the camera view to wider angle perspectives. The initial prototype included a single 5 by 4 inch mirror placed at 90 degrees (perpendicular) to the PiCamera lens, but the final stereo view was angled and warped. The final prototype included two 3 by 3 inch mirrors connected at 170 degrees, placed directly in front of the single PiCamera lens. This double mirror configuration fixed the warped perspective and allowed the reflected image to be a close replica of the original image. This configuration also allowed for a small amount of disparity between the images, which allows for the benefits of stereo camera systems to be utilized in an optimal fashion. In figure 15 the results can be seen. The figure is broken into three parts: (a) is for the final prototype configuration, (b) is for the theoretical application, and (c) is for the final original and reflected image outcome. The final results are shown below.

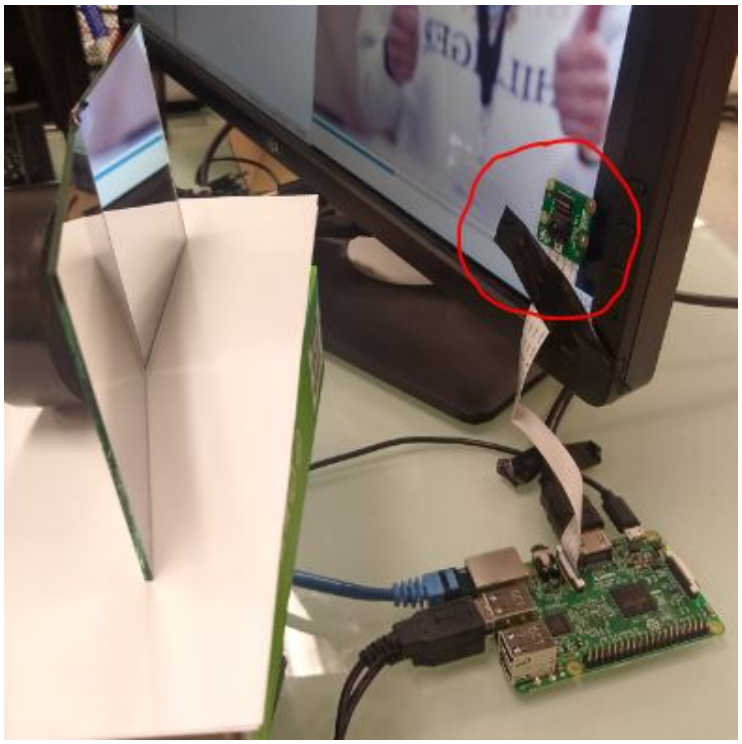


Figure 15 (a). Stereo mirror cam setup

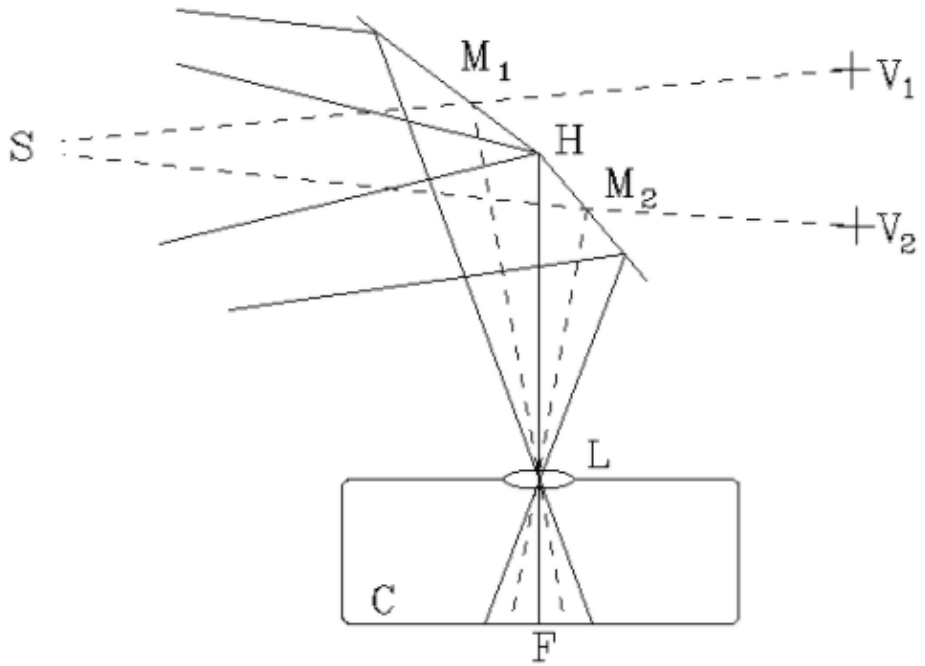


Figure 15 (b). Mirror Schematic (https://www.lockhaven.edu/~dsimanek/3d/stereo/3dgallery16.htm)

Figure 15 (c). Stereo mirror cam output

Patent and Standards Research Related to Design (CLO 6):

The People Location Tracking & Analysis team has worked on utilizing open-source research, which helped the team in implementing the system and improving it with different sides. A significant portion of the project has been completed with the OpenCV library, and much of the development of the algorithms for social distance and location tracking were initialized with the PyImageSearch website and provided tutorials. However, none of the references used combined all the traits together into a single system. Person Detection and transforms are not new, however from our research there haven't been any systems that used the detection of persons to map their location onto a floorplan in a building. The closest we found was a research paper that implemented computer vision and a step counter to assist vision impaired people in navigating indoors. So, there is the possibility that this could become a patentable technology.

Proof of Concept (CLO 8):

The team has collected most of the footage needed to run our detection algorithms on. The footage contains several hours of vacant rooms in between classes being held, so we are in the process of trimming that footage down to 2-5 minute usable sections for testing and modifying of our detection algorithms and 2D mapping. At the time we were using wi-fi command signals to start recordings and ran into the problem of not having synchronized video between the cameras in a room. We are developing a program to crop the videos by the frame offset between them. The room to pixel point maps are being generated to get exact person locations from our footage, and this will help with the 2D mapping we will do, but our code got deleted so we are in the process of rewriting it. We are using Node-RED to connect and visually represent our system flow and connectivity and we are using mosquito to get perfectly synced video streams that are saved on the monster server which helps us sync the streams and timing perfectly. Our working code last semester showed that the concept worked but was highly inaccurate and we believe that with improved depth information and more precise pixel locations for our 3D to 2D transformation we will have a better end product.

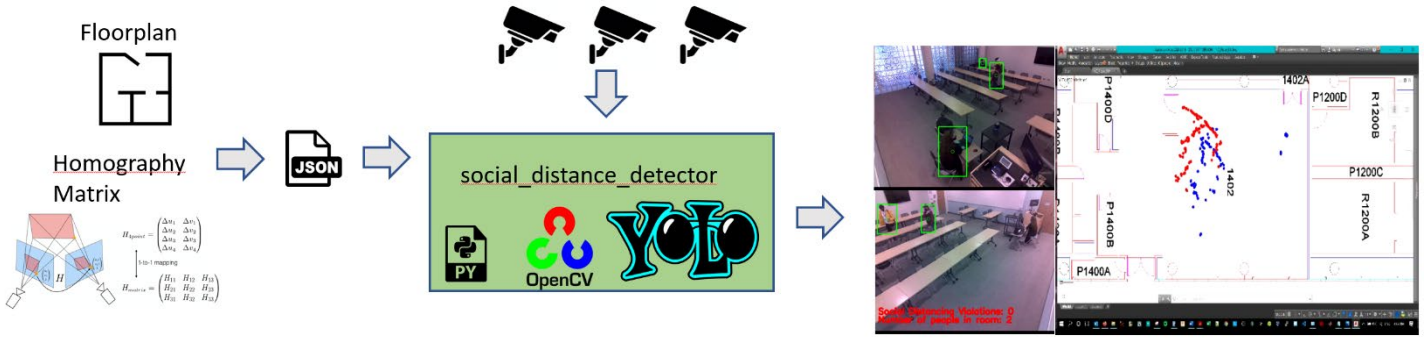


Figure 16. End product flow map

List of Project Design and Implementations Task/Responsibilities:

Stakeholders/Interviews:

1. Chief Phibbs: Amanda (main) and Team
2. Cary Weatherford: Patrick (main) and Team
3. AFRL: Team

Hardware: Circuit Design:

1. Receiver/Transmitter Circuit: Jonathan (main) and Patrick

Device Implementation:

1. Building Camera Modules: Patrick (main) / Amanda / Chance / Huseen / Fadhel
2. Building Camera Mounts: Patrick (main) / Chance / Huseen / Amanda / Fadhel
3. Placing Camera Mounts: Patrick / Chance / Huseen / Amanda / Fadhel
4. ESL Filming Permissions: Huseen (main) and Patrick

Software:

1. Node Red communication Protocol: Patrick
2. Mirror Stereo System: Amanda
3. Mapping Algorithm: Chance (main) / Amanda
4. Training own Neural Network: Patrick

Deliverables:

1. PowerPoints/Presentation: Huseen (main) / Team
2. Status Reports/Poster: Huseen (main) / Team
3. Video: Team