# Spy Camera Localization Midterm Report

Gustavo Fonseca*
Ansh Kaushik*
gvf3@illinois.edu
anshulk3@illinois.edu

## ABSTRACT

The goal we set out to accomplish in this midterm was to try to localize a hidden camera or transmitter that was hidden within a certain area. We would try to localize it by approximating our relative position with a Inertial Measurement Unit (IMU) connected to our raspberry pi, and then merging this with the Received Signal Strength Indicator (RSSI) values we received while walking through the course.

## 1 LOCALIZATION SYSTEM

In this section we will outline each step within our pipeline to determine our approximate location of the camera or transmitter.

### 1.1 Preprocessing

When collecting data we would record the Accelerometer's XYZ acceleration values and the Gyroscope's XYZ angular acceleration data with a timestamp for each data-point. We will be using Accelerometer's Z axis and Gyroscope's Z axis to determine step detection and angle of movement.

*1.1.1 Data Calibration.* After the data collection is done, we calibrate the data with an area where the data is not changing, indicating that person is not moving. We always start any walk with large amount of time where we stand still, thus we can always use the start of the CSV data to calibrate the data.

*1.1.2 Step Detection.* After calibration, we smooth out the Z acceleration by apply a savgol filter with 21 taps and a polyorder of 2. We chose these values as they were a good middle ground as it eliminated most of the noise but still retained the spikes caused by our steps. We then found the peaks of the smoothed Z data, using a threshold of 0.01 and a distance between peaks of 3. We chose this threshold as this removed most peaks that were noise and retained all peaks we believed to be representing steps. Each of these peaks represents one step taken, so it was very important that the number of peaks and steps were the same.

*1.1.3 Walking Direction.* To determine the direction we are walking in, we take the integration of the Gyroscope's Z axis, which gives us the Yaw, the current angle the Raspberry Pi is facing in the XY plane. This angle allows us to calculate our X and Y positions more accurately as we can determine current walking direction by Eq. 1, where $\theta$ the current angle we are facing and $\psi$ is the yaw.

$$\theta = [\cos\psi, \sin\psi] \tag{1}$$

*1.1.4 Trajectory.* To determine the trajectory of motion we will combine the step detection with the walking direction calculated above. To do this we calculated the average step size of a 6 foot male, and change the position by this much based on the calculated angle when a step is taken. We can than plot the trajectory of our walking pattern by plotting where we are after each step.

*1.1.5 RSSI.* We needed to filter out all packets that did not have the correct MAC Address. To solve this, we used a WireShark Filter to only record the RSSI values of the MAC address we were given. We then saved the timestamp that the packet was received.

### 1.2 IMU and RSSI Merging Method

We created our RSSI and IMU data with 2 separate CSV's. Thus, we needed to merge the data together to create an accurate approximation of the trasmitter's location. An important thing to consider is that the RSSI CSV file had a lot less data as the polling rate was much smaller than the IMU data. To solve this, we merged the data together based on the timestamps, where we nade a new column in the IMU CSV data that held the RSSI value at that timestamp, if the timestamps of the 2 CSV's were not the same value, then we match the RSSI Timestamp to the nearest IMU timestamp. At this point, there are still many timestamps within the IMU data that have no RSSI value, to solve this, we propagate the last filled RSSI value into each empty RSSI data until another filled RSSI value. See Figure 1 for an example merge.

As a bit of a side note, this merging strategy actually saved our Day 2 results, as we accidentally appended all the Day 2 RSSI data onto the end of our Day 1 RSSI CSV. We originally thought that it had overridden the data, and we analyzed the data for our Day 2 estimate. After the competition, we realized that this had happened and that our trajectory plot and interpolation plot for Day 1 were identical to the original plots from day 1 using this CSV. Because we merged the data based on the timestamp, we essentially isolated the RSSI data since the timestamps of the IMU data of the different days only merged with the specific days RSSI data.

| IMU Data | | RSSI Data | | | Merged Data | | |
|---|---|---|---|---|---|---|---|
| **TS** | **Data** | **TS** | **RSSI** | **merge** | **TS** | **Data** | **RSSI** |
| 10 | $(x_1, y_1, z_1)$ | 11 | -40 | $\longrightarrow$ | 10 | $(x_1, y_1, z_1)$ | -40 |
| 13 | $(x_2, y_2, z_2)$ | 15 | -50 | | 13 | $(x_2, y_2, z_2)$ | -40 |
| 14.5 | $(x_3, y_3, z_3)$ | | | | 14.5 | $(x_3, y_3, z_3)$ | -50 |

**Figure 1: Merged CSV Example**

### 1.3 Walking Strategy

We had different walking strategies for each day. In this section we will discuss the strategy before and during the walking for each day. To minimize angle error, we decided that would only turn in 90 degree intervals to be able to section the data easier. To give us real time RSSI updates we programmed the LEDs on the SenseHat to

*Both authors contributed equally to this research.

turn on and off depending on the average of the last 3 RSSI values, with different patterns for every 2.5 dB change from -45 dBs and -60 dBs.

*1.3.1 Day 1.* In the first day, we tried to cover every corner of the area. We wanted to make sure that we were close to the transmitter at least once. Once we walked around most of the testing area we then tried to go back to areas that had a RSSI values while walking. Figure 2 Shows our path for day 1. We can see that when we reached top intersection between Zone 3 and 4 we chose to around Zone 3 as we thought that the RSSI values there were stronger. Knowing that the transmitter was really in Zone 4, this is a critical mistake. This inaccuracy in RSSI values may have been due to the delay in the change in the LEDs, so when we saw high values in Zone 3 originally they may have have still been from Zone 4. Finally, we saw a large spike around Zone 1 when originally going through each zone, so when in the center of each zone we decided to go down towards Zone 1 again. Again, this was a mistake, we should have covered each possible path to gather new data. One theory for why the values we received were strong in Zone 1 was that the path from Zone 4 to 1 was not very obstructed, compared to Zones 2 and 3 where there was monitors and other large objects that could obstruct the signal.

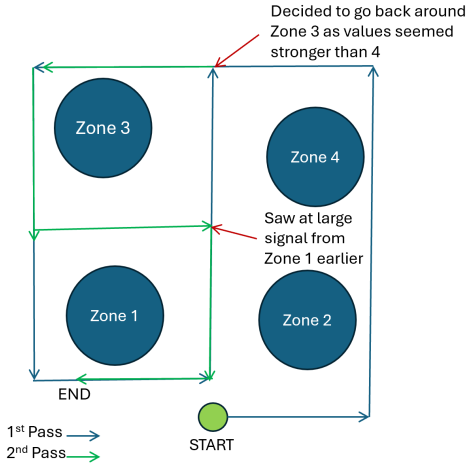


**Figure 2: Day 1 Walking Path**

*1.3.2 Day 2.* In the second day, we again tried to cover every section of the area to find the camera. The main difference this day was that once we got a signal from the camera we would adjust our walking pattern to try to find the camera. Since we were not always receiving signals from the camera we set up a system to clear the LEDs every 0.5 seconds when walking. We can see that in Figure 7 the first time we received a signal was around zone 3, so we thought that the camera may have been on Zone 1 and pointing to Zone 3, thus we deviated from original path of going around Zone 1 and went through the intersection between Zone 1 and 3. When going through this area, we received a very strong RSSI reading, signaling to us at least, that camera was nearby. We then proceeded to walk around the center to try to pin point the camera. We walked near Zone 2 and 4 to see if we received a signal and got no response, further solidifying that the camera was in Zone 1.

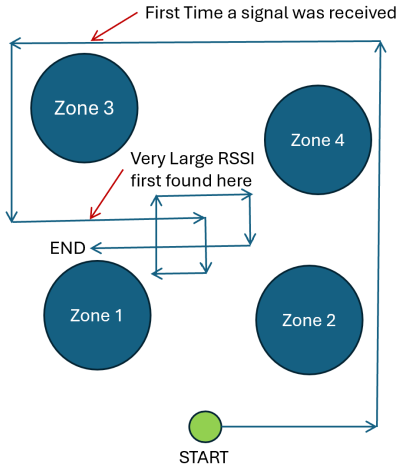


**Figure 3: Day 2 Walking Path**

## 1.4 Localization Algorithm

Once the data is merged together we can then go about looking for the maximum RSSI value. We took an average of 5 values ahead and 5 values behind the current position into account for finding the maximum value. This would be our estimated position based on the raw data. Of course, the transmitter is not exactly on the point we measured. To counteract this, we created a 2D interpolation plot to try to find the targets exact position if we went around the exact location. We used a linear method of interpolation as it would make sense for this application. We then took the maximum value from this plot to use as our estimated position for this plot.

## 1.5 Post-Processing

Once we got the data recorded, we started to look at the data. One issue that was our scale of steps was not consistent with the scale of room and what our actual path was. We corrected this by scaling down the trajectory plot with the known dimensions of the competition area. We also considered that sometimes our steps were not the same size every time. A key example of this was when we got a strong signal from the camera in Day 2, After this in order to improve accuracy of readings we took very small steps around the original point. To combat this, we changed the step size of each section where we felt that the original step size was not accurately representing our movement. We only thought of this during the data processing of Day 2, so we only implemented this for the Day 1 data for this report below.

Another issue was that the angles we got were not returning exactly 90 degrees after integration. Thus, we decided that the best course of action would be to round the yaw data to the nearest 90, such that our turns had perfect accuracy. This change allowed us to get more accurate data from our IMU data.

In the final step, we would compare the max average RSSI value found from the trajectory plot and the interpolation plot. Which ever one made more sense in terms of real life accuracy we used for our final position estimation. We used the interpolation plot point for Day 1 and the trajectory plot point for Day 2.

# 2 DATA BENCHMARKING

## 2.1 Day 1

For visual displays in the first day, we decided to have a trajectory plot where the color of the positions correspond to RSSI, and then have a interpolation plot where a lighter color also corresponds to a higher RSSI. For the trajectory, we used a rounding system to round all of the radial values of the yaw to the nearest Pi/2 value. This ensured that all of our 90 degree turns were recorded as 90 degrees to ensure optimal accuracy. In both plots we marked the highest RSSI value, based on getting the highest average on 5 consecutive values, with a "+" and a red dot respectively.

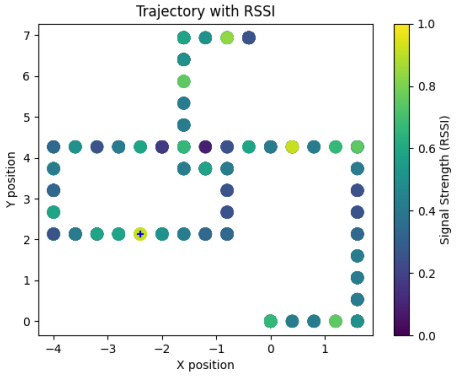


**Figure 4: Day 1 Trajectory Interpolation**

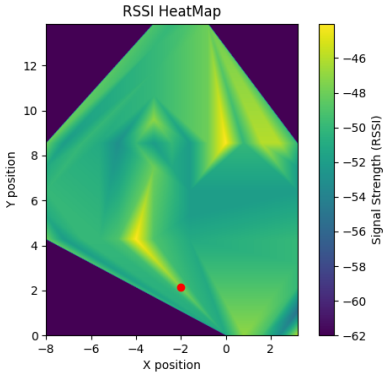


**Figure 5: Day 1 HeatMap Interpolation**

## 2.2 Day 2

For visual displays in the second day, we decided to have a trajectory plot where the color of the positions correspond to RSSI, and then have a interpolation plot where a lighter color also corresponds to a higher RSSI. As we did in day 1, we rounded the radial yaw values to the nearest Pi/2 radians to ensure our 90 degree turns remainied at 90 degrees. However, we utilized scaling of steps to account for the fact that the step length in reality decreased significantly after halftime. This helped the location accuracy be much higher as compared to day 1. In both plots we marked the highest RSSI value, based on getting the highest average on 5 consecutive values, with a "+" and a red dot respectively.

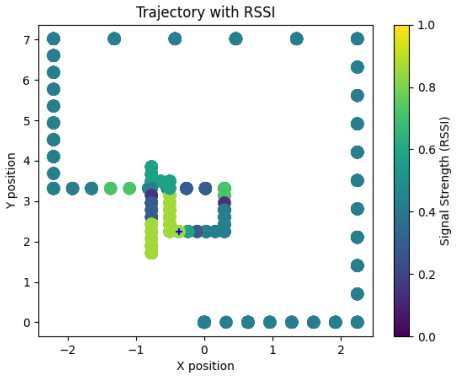


**Figure 6: Day 2 Trajectory Interpolation**

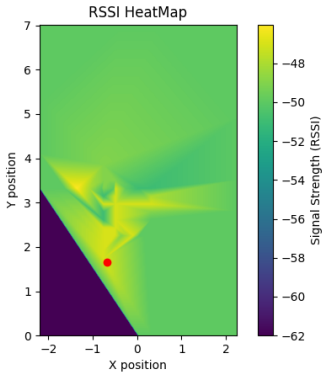


**Figure 7: Day 2 HeatMap Interpolation**

# 3 EVALUATION

## 3.1 Day 1

In day 1, we can see that the trajectory plot was not accurate to the actual distances traveled in our trajectory. Although the number of turns to the left and right is accurate, the distance traveled was so inaccurate, it is hard to analyze it and get an answer even close to

correct. Since we had each step as the same distance, it resulted in any change to stride length having huge impacts. The trajectory plot ends at the highest y value even though our actual path traveled has the ending point closer to the starting y value than the max possible.

The correct answer was zone 3 in the top right quadrant whereas we had our guess in zone 1 in the bottom left quadrant. Something of note is that we could see a large light green chunk in the top right quadrant of our RSSI HeatMap but the largest average value was not there. Perhaps with a better trajectory plot, the values there would be more accurate.

### 3.2 Day 2

In day 2, due to our scaling of each step using context resulted in a much more accurate trajectory plot compared to the path. Because of this, our guess was in zone 1 which is correct to where the actual camera was. Given how many short steps we took once we got to the center of the map, we essentially massively reduced the step length constant after the timestamp where we entered the center to account for this.

If we look at the heatmap, we can see a clear spot of light green in zone 1 which is where the camera was located. Additionally, because our trajectory plot had correct distance, we could more accurately use the heatmap to interpolate away from the trajectory and get closer to the potential value of the camera.

## 4 CONCLUSION

Overall, what we noticed is that having each step be used as the same distance is fundamentally flawed as there is simply no realistic way for use to consistently have each step we actually take be the same distance. We have to account for turns, indecisiveness, stumbles, and fluctuations in data collection. This is why writing down the path helps a lot because we could use it to contextualize and appropriately scale the step length to account for a change in pace.

We also saw in day 1 that if there are two areas where high RSSI values are detected, the incorrect one may be picked if it has a randomly higher RSSI value to artificially boost the average for that specific location. Perhaps in the future, implementing outlier adjustments could help us truly grasp the area with strong RSSI values all around instead of an area which got one super high value that made the running average seem larger than what it actually should've been. Signal strength is extremely shakey especially with a camera with many objects obstructing its view and signals sent.

Our adjustments for day 2 resulted in a much more accurate guess and a more natural looking trajectory plot. This is all despite getting less packets from the camera and having to do a much less simple trajectory to ensure we got the limited packets. Using the combination of step counting, rounding, scaling, and interpolating, we were able to map out a general plot of where to expect a strong signal for the camrea.

## 5 GITLAB LINK

Link: https://gitlab.engr.illinois.edu/cs437fa24group10/midterm1/-/tree/main