

Department of Electrical, Computer, and Software Engineering

**Part IV Research
Project**

Final Report

Project Number: 16

Occupancy detection system
for indoor environments

Hasnain Juzar Cheena

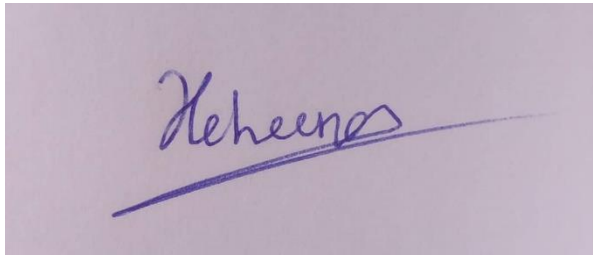
Abin Thomas

Prof. Sing Kiong Nguang

Date 29/09/2019

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

A handwritten signature in blue ink, reading "Heheena", with a long horizontal line extending to the right.

Name: Hasnain Cheena

ABSTRACT

The global increase in electricity demand has raised the need to develop accurate and robust occupancy estimation systems. The integration of occupancy detection systems with building services (e.g. HVAC and lighting) is an opportunity to improve the electricity usage of buildings and other indoor spaces. In this study, we propose an accurate and portable real-time occupancy detection system for indoor spaces. The system is based on millimetre-wave radar because of its non-intrusive characteristics and high accuracy. Moreover, the system was developed on the Raspberry Pi 4 for scalability and portability purposes. The software pipeline consists of a density-based clustering algorithm to identify people, coupled with a Kalman filter to track people. The proposed system has been evaluated in various room settings and scenarios. Experimental results were compared to the commercially available Texas Instruments occupancy detection system. The experimental accuracy of the proposed system ranged from 98% for one person to 75% for five people with an average position error of 0.32 metres and 0.29 metres in the x-y direction, respectively. In comparison, the Texas Instruments system had an experimental accuracy ranging from 94% for one person to 43% for five people with an average position error of 0.33 metres and 0.3 metres in the x-y direction, respectively.

TABLE OF CONTENTS

1. Introduction	1
1.1. Division of Work.....	2
2. Literature Review	2
2.1. Camera	2
2.2. Ultrasonic	2
2.3. Passive Infrared Sensor (PIR)	3
2.4. Radio Frequency Identification (RFID)	3
2.5. Hybrid Systems	3
2.6. Millimeter Wave Radar (mmWave).....	4
3. Proposed System.....	5
3.1. Hardware.....	5
3.2. Software	6
3.2.1. Parsing	6
3.2.2. Noise Reduction.....	6
3.2.3. Clustering.....	7
3.2.4. Tracking.....	8
3.2.5. Graphing	10
4. Results & Discussions	10
4.1. Algorithm Optimization	10
4.1.1. Zone SNR Thresholds.....	10
4.1.2. Clustering Parameter Optimization.....	11
4.1.3. Kalman Filter Parameter Optimization	12
4.2. Embedded Implementation Considerations.....	13
4.3. Testing.....	14
4.4. Results	14
4.4.1. Occupancy Estimation Accuracy	15
4.4.2. Tracking Accuracy.....	16
4.4.3. Timing Analysis.....	17
5. Conclusions & Future Directions.....	17
Acknowledgements.....	18
References	18
Appendix A: Kalman Filter Equations	19
Appendix B: SNR Filtering Model	20
Appendix C: Accuracy Graphs	20

TABLE OF FIGURES

Figure 1: Flow of information	5
Figure 2: Sensor constraints.....	5
Figure 3: Software Pipeline	6
Figure 4: Demonstration of SNR Filtering	7
Figure 5: Pseudocode for Clustering.....	7
Figure 6: Density variations as a person moves around.....	8
Figure 7: GNN Diagram and Pseudocode	9
Figure 8: Graphical interface showing location and occupancy estimate	10
Figure 9: Model summary and model fitted on data	11
Figure 10: 2 views of the 3D Scatterplots showing parameter set that maximized accuracy	12
Figure 11: Kalman Filter smoothing characteristic curves	13
Figure 12: Performance gains with/without primary noise removal	14
Figure 13: Various testing scenarios and experimental setup	14
Figure 14: Overall accuracy and Per Group accuracy results	15
Figure 15: Graphs showing positional error in Y and X direction	16
Figure 16: Graph of time per frame (ms) versus number of people.....	17
Figure 18: Accuracy graph over all datasets for our system versus TI	20

TABLE OF TABLES

Table 1: Range with corresponding SNR thresholds	6
Table 2: Ground truth for positional testing	16

1. Introduction

The rapid spread of the Internet of Things has brought about an interest in developing smart systems that use energy more efficiently. Such systems are more critical than ever, with studies projecting that global energy usage will have increased by 50% in 2035 [1]. An opportunity exists for improving the energy consumption of buildings and other indoor spaces with the application of occupancy detection systems. Studies show that occupancy detection can reduce energy usage for lighting and HVAC systems by between 25% to 80% [1,2]. Places where building automation systems have been deployed, reported a reduction in electricity usage by around 15% [3]. Additionally, these systems could be of use in security applications. First responders would benefit from occupancy information in events such as fire rescue, collapsed buildings and hostage scenarios [4]. Occupancy detection systems could also help commercial businesses make optimal decisions when it comes to staff management. For example, based on occupancy information, managers could schedule employee shifts according to shopper traffic, ensuring the business is not over/understaffed [5].

Ongoing research in occupancy detection employs a variety of sensing approaches and algorithms. Researchers typically use sensor technologies such as passive infrared sensors (PIR), ultrasonic sensors, millimetre-wave radar and digital cameras. These sensing technologies are coupled with numerous algorithms to process the sensor signals. The algorithms typically involve classification via behaviour/activity analysis or pattern detection. Additionally, tracking algorithms are added onto the classifiers to increase the accuracy of occupancy estimates.

This study focuses on developing accurate and portable occupancy detection for real-time application in indoor locations. Our proposed system uses millimetre-wave radar (IWR1642BOOST sensor) because of its non-intrusive nature, ability to detect near-stationary/stationary people and high accuracy. Furthermore, the system will operate on an embedded platform, the Raspberry Pi 4, creating computing constraints to introduce a portability and scalability aspect. The experimental results of our proposed system are compared to the commercially available system from Texas Instruments.

Section 2 contains literature review which examines works related to indoor occupancy detection. Section 3 outlines the design of our system and Section 4 presents testing methodology, results and subsequent discussions. Finally, Section 5 contains the conclusions drawn from our research.

1.1. Division of Work

Abin and I together worked on setting up and parsing the information sent from millimetre-wave sensor. After that stage, I primarily worked on the clustering/filtering while Abin worked on the tracking. Then together we concatenated all the processing stages as one whole pipeline.

2. Literature Review

This section outlines past work related to indoor occupancy detection.

2.1. Camera

Vision technology such as cameras take digital images of the scene, processing them to extract occupancy information. Studies [6] and [7] present indoor occupancy detection using camera technology. In these studies, machine learning methods are employed to detect occupancy. These methods include decision trees, hidden Markov models and convolutional neural networks. Machine learning is a computationally-intensive process and therefore, cannot readily be implemented onto an embedded system. Moreover, typically, camera-based occupancy systems require a clear view and the right lighting conditions. This is displayed in [8], where the system uses background subtraction and the Lucas & Kande tracking algorithm to determine indoor occupancy. The system had an experimental accuracy of 97% under lab conditions, but the accuracy during field-testing dropped significantly. Additionally, another critical problem with camera systems is their intrusive nature, leading to privacy concerns. However, research is being done to discover whether it is possible to use lower resolution cameras to circumvent privacy issues [9].

2.2. Ultrasonic

Ultrasonic sensors measure occupancy by examining the returned echo signals [4]. Recent work in [10] proposed an occupancy estimation system using multiple ultrasonic sensors. The system used the signal-to-noise ratio (SNR) of a returned ultrasonic pulse as an indication of occupancy. A custom tracking algorithm

was added onto the software pipeline to increase accuracy. The experimental accuracy results were around 80%.

Further work done in [4] proposed a pipeline in which ultrasonic sensor data was processed using density-based clustering (DBSCAN) followed by a regression model. The system was able to estimate occupancy with approximately 10% error on a wide variety of room sizes. Both studies show that ultrasonic-based occupancy systems can perform with high accuracy. However, ultrasonic-based detection is plagued by significant safety concerns. Ultrasound waves affect people wearing hearing-aids and can be heard by a variety of animals, rendering ultrasonic based occupancy detection unsuitable for indoor environments [1].

2.3. Passive Infrared Sensor (PIR)

PIR sensors detect the change in heat radiation caused by the movement of occupants [1]. This information is then used to infer the occupancy within a space. PIR sensors are commonly used in both commercial and academic systems because of their low cost and widespread availability [11]. However, a significant disadvantage of PIR sensors is that inherently, they cannot identify people that are motionless/near-motionless [1]. [3] aimed to overcome this by using multiple PIR sensors coupled with particle filtering to improve accuracy. The experimental accuracy of the system varied between 29.6% to 98.3%.

2.4. Radio Frequency Identification (RFID)

RFID technology requires the occupants of an indoor space to carry RFID tags which communicate with RFID readers. The occupancy information is derived from the communication logs of the RFID readers [12]. Studies from the University of Southern California [13] and Tsinghua University [14] show that RFID-based occupancy detection systems have high accuracy at room-level ranging between 62% to 93%. However, RFID-based occupancy systems require that occupants wear a tag, and enough readers must be installed.

2.5. Hybrid Systems

A large proportion of ongoing research employs hybrid systems which combine multiple sensor technologies to overcome the limitations of the individual sensors alone. Studies [15], [16] and [17] prove that high accuracy can be achieved when integrating multiple sensor types. However, systems with various sensing technologies

require more computational power to process the additional signals. Thus, it may not be feasible to use multiple sensing techniques in an embedded implementation.

2.6. Millimeter Wave Radar (mmWave)

Radar technology, such as millimetre-wave sensors examine the modulated backscatter of electromagnetic signals to determine occupancy [1]. Millimetre-wave technology is new and thus is being investigated widely by both academic and commercial institutions. Infineon Technologies conducted a study using cardiopulmonary data gathered from millimetre-wave sensors to determine occupancy [2]. Doppler information was extracted from the millimetre-wave signals. This doppler information was fed through bandpass filters to obtain the required cardiopulmonary data. The experimental accuracy of the system was approximately around 90% between 1 and 3 people. A complementary study conducted by the University of Hawaii used millimetre-wave radar and an equivalent filtering technique produced similar results [18]. Furthermore, in [19] density-based clustering (DBSCAN) along with a Kalman filter was used to track the movements of people. The Kalman filter was coupled with the Hungarian algorithm for data association to extend the system to multiple target tracking. The experimental accuracy of this system was 89% for up to 12 people with a median position error of 0.16 meters. The study noted that DBSCAN was challenging to use because of the varying density nature of the clusters. Moreover, the Hungarian algorithm would have caused bottlenecks in the system when higher numbers of people were present because of its algorithmic complexity. Thus, making it unsuitable for a real-time embedded application.

Additionally, Texas Instruments (TI) built an occupant counting system centred around their IWR1642BOOST millimetre-wave sensor [20]. The system employs density-based clustering (DBSCAN) with an extended Kalman Filter (EKF). The system had a reported accuracy of 51% to 99% between 1 to 5 people. Texas Instruments employed an EKF because their proposed system converted the polar radar measurement to cartesian. The conversion was performed for ease of use but in turn, created additional computational load.

3. Proposed System

This section outlines details of the hardware and software implemented within the proposed system.

3.1. Hardware

The embedded system consists of a Raspberry Pi 4 (1.5GHz, 4GB RAM), the millimetre-wave radar sensor (IWR1642BOOST) and a monitor. The flow of information is shown in Figure 1. The sensor emits a radar signal, taking a snapshot of the indoor location at a given point in time. The returned radar signal undergoes preliminary processing on the sensor, the output of which is a point cloud. This point cloud is a collection of points that represents detected people. The point cloud is then processed on the Raspberry Pi 4. The output of the processing is information on identified targets, which is then displayed on a monitor.

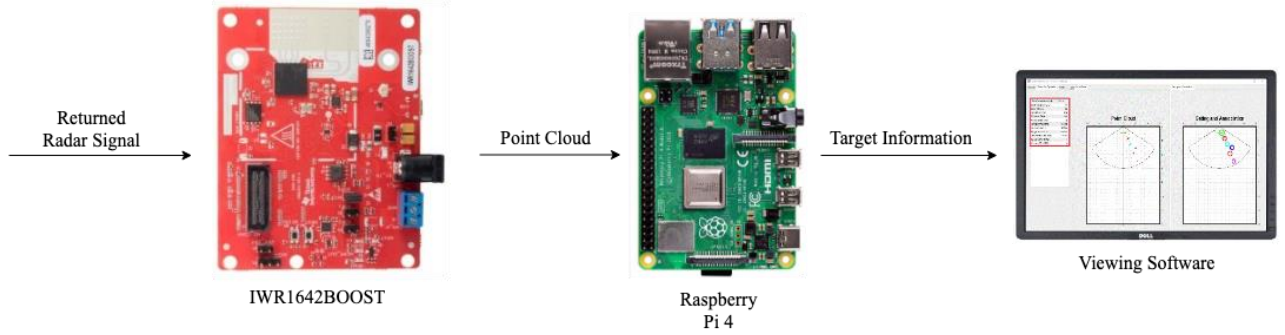


Figure 1: Flow of information

The configuration parameters for the sensor used in this study are: A range (r) of 1 to 6 metres and an azimuth (ϕ) of -60° to 60° . The sensor constraints are shown in Figure 2.

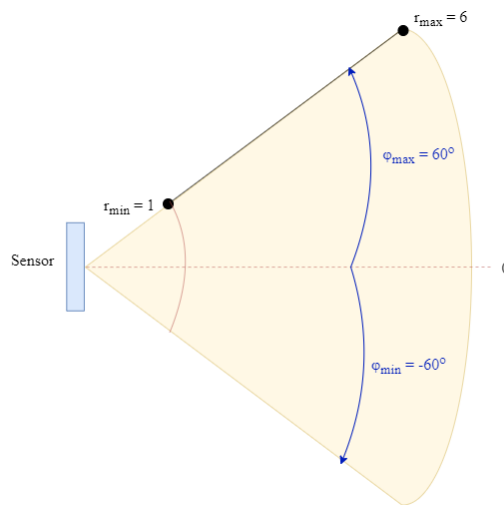


Figure 2: Sensor constraints

3.2. Software

The processing stack is partitioned into five distinct stages; Parsing, Noise Reduction, Clustering, Tracking and Graphing.

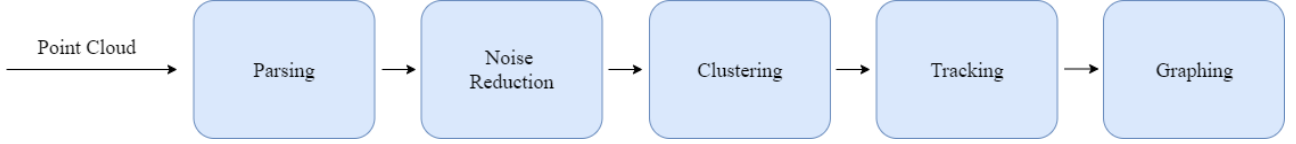


Figure 3: Software Pipeline

3.2.1. Parsing

The millimetre-wave sensor transmits the point cloud data to the Raspberry Pi. This information is structured as follows: Each frame has a header, followed by segments containing the point cloud information encoded in type-length-value (TLV) format. Further information can be found in the sensor manual [20].

3.2.2. Noise Reduction

The noise reduction stage is further divided into preliminary noise removal and signal-to-noise ratio (SNR) filtering. Preliminary noise removal discards clusters that are identified to be noise. These are clusters of points that are too small to represent people. Removal of these clusters improves performance. The impact of preliminary noise removal on performance is further discussed in Section 4.2.

SNR filtering is the second stage of noise reduction. The higher the SNR of a point, the higher certainty that the point corresponds to a person. SNR filtering aims to reduce the size of the remaining clusters to further improve performance and create additional distance between occupants. The impact of SNR filtering is documented in Figure 4. Our system performed zone-based SNR filtering. The range of the sensor was divided into three regions. Observations in each region were filtered using a different SNR threshold. Table 1 contains the ranges and the associated SNR thresholds.

Table 1: Range with corresponding SNR thresholds

Range	SNR Threshold
1-3m	338
3-5m	139
5-6m	53

Using different SNR thresholds was required as, on average, the SNR of observed people increases with range. This is discussed in more depth in Section 4.1.1.

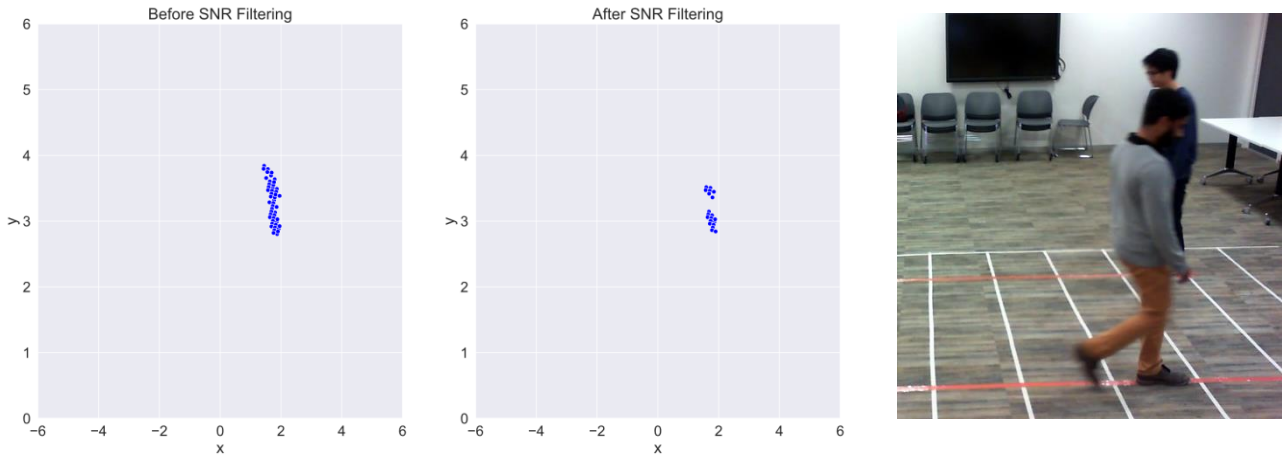


Figure 4: Demonstration of SNR Filtering

3.2.3. Clustering

The clustering stage aims to identify the number of people in the room. The system employs a density-based clustering algorithm to figure out how many people are within the space at a given time. The general pseudocode to the algorithm is shown in Figure 5.

Algorithm	Clustering (frame, minimumClusterSize, maximumDistance)
1.	Create a graph where each node is a point and each weighted edge is the distance between the points
2.	Remove edges on graph that are greater than the maximum allowable distance (maximumDistance)
3.	Use depth first search algorithm to extract connected components (clusters)
4.	Remove clusters that are smaller than the minimum allowable cluster size (minimumClusterSize)
5.	Use Kmeans algorithm to extract the centroid of each cluster

Figure 5: Pseudocode for Clustering

The parameters associated with the clustering algorithm are:

- Minimum cluster size (minClusterSize): smallest number of points required to classify as a cluster.
- Maximum distance between points (maxDistance): largest distance possible between neighboring points to be associated within the same cluster,

The Texas Instruments system uses the famous density-based spatial clustering for applications with noise algorithm (DBSCAN). However, DBSCAN is not ideal because of the nature of the clustering data. DBSCAN

assumes a constant cluster density which is not the case with our data [21,22]. As shown in Figure 6, a person closer to the sensor (located at origin) is represented by a denser and more uniform cluster. In contrast, a person that is further away is represented by a less dense and more variable cluster.



Figure 6: Density variations as a person moves around

The density-based clustering algorithm we designed for this task can manage variable cluster densities. However, the limitation of this algorithm is it cannot handle noise as well as DBSCAN. Thus, creating the need for the prior noise reduction stage. Other algorithms like HDBSCAN [21] and VDBSCAN [22] can handle varying density clusters but were too computationally expensive for a real-time embedded application.

3.2.4. Tracking

The tracking stage is required to locate people as they move through the indoor space. In this stage, a recursive Kalman filter (RKF) is employed. The purpose of utilising the RKF is to identify people in the presence of missing or erroneous data occurring when motionless behaviour is exhibited (e.g. people sit down or stand-still). The Kalman filter is able to deal with error-prone/missing data because it can estimate measurable and unmeasurable system states while smoothing out both the system/measurement noise.

The RKF is an extension of the Kalman filter where the error covariance and Kalman gain are recalculated at each iteration. Recalculating the Kalman gain and error covariance gives the estimator practical flexibility; the versatile ability to adjust to different types of behaviour (i.e. running, walking), increasing tracking reliability.

A constant velocity model was applied. Δt was set to 50 milliseconds which is the frame rate of the millimetre-wave sensor. The state space model of the system is outlined in equation 1, 2 and 3:

$$x[k] = \begin{bmatrix} r[k] \\ \dot{r}[k] \\ \theta[k] \\ \dot{\theta}[k] \end{bmatrix} \quad (1)$$

$$x[k+1] = Ax[k] + Q \quad (2)$$

$$y[k] = Cx[k] + R \quad (3)$$

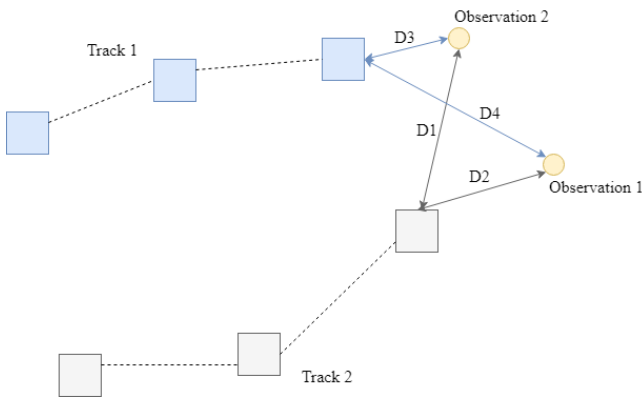
$$\text{Where } A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Q, R are the system noise covariance and measurement noise covariance respectively

The base Kalman filter algorithm consists of two steps; prediction and update. The prediction step estimates the next location of a person using the previous location. The update step fuses the observation with the predicted value as well as reducing both the measurement and system noise. Refer to Appendix A for the predict and update step equations.

The recursive Kalman filter is only able to track a single person at a time. Therefore, a data association method to match people between discrete frames of information is required to enable multiple target tracking.

The global nearest neighbour (GNN) data association method was used in the proposed system. The pseudocode for this method is shown below in Figure 7.



Algorithm Global Nearest Neighbour (pastCentroids, newObservations)

1. Calculate euclidean distances between pastCentroids and newObservations (i.e. Find D1, D2, D3 and D4 distance between the points).
2. Find the shortest distance from the calculated distances (i.e. D3 is shortest distance).
3. Associate the centroid and observation linked by this distance (i.e. associate Track 1 and Observation 2).
4. Repeat Step 2 and 3 for all unassociated centroids and observations (i.e. associate Track 2 and Observation 1).

Figure 7: GNN Diagram and Pseudocode

After GNN is performed, the associated centroids can be passed through the update step of the Kalman filter, producing location estimates of each occupant.

The TI system uses an extended Kalman filter (EKF). The EKF is a tracking algorithm for non-linear systems. Our proposed system operates solely in the polar coordinate domain. In comparison, TI converts the polar coordinates to cartesian for tracking. This is an unnecessary addition of computational load.

3.2.5. Graphing

The scientific plotting library *pyqtgraph* [23] was used to display the number of people and their locations within the indoor setting. Figure 8 shows the graphical interface that was shown to users.

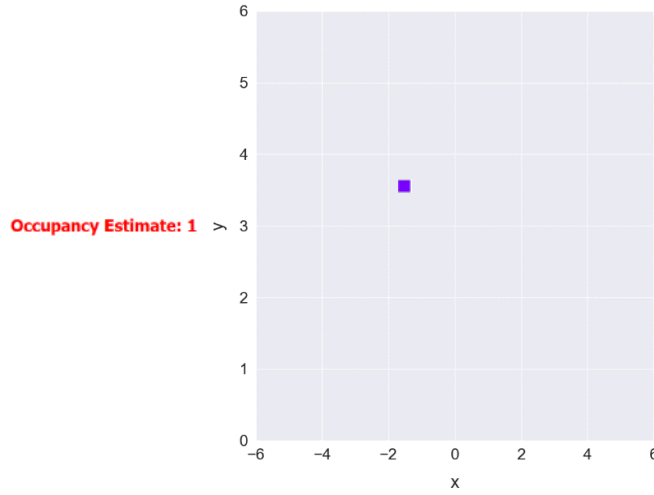


Figure 8: Graphical interface showing location and occupancy estimate

4. Results & Discussions

This section contains details on the optimization of parameters, testing methodology and experimental results.

4.1. Algorithm Optimization

This section outlines the optimization procedures for SNR Filtering, Clustering and Tracking.

4.1.1. Zone SNR Thresholds

There was strong evidence that a multiplicative linear model can model the relationship between range and SNR as the p-value of the model was significant at a 5% level (highlighted in Figure 9).

```
Call:
lm(formula = log(SNR) ~ Range, data = df)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.39508    0.06819   93.78  <2e-16
Range       -0.47187    0.01834  -25.73  <2e-16
```

	fit	lwr	upr
373.6093	338.12807	412.81382	
145.3970	139.25725	151.80734	
56.5839	53.06545	60.33563	

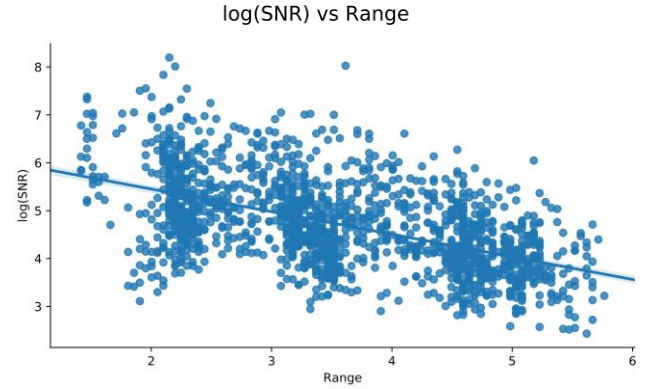


Figure 9: Model summary and model fitted on data

The 95% confidence intervals at the boundaries of the zones (ranges of 1m, 3m and 5m) were evaluated (shown in Figure 9). We estimate that on average:

- A point at 1m range will have a SNR between 338 and 413
- A point at 3m range will have a SNR between 139 and 152
- A point at 5m range will have a SNR between 53 and 60

The lower bound of the confidence intervals were used as the SNR filtering thresholds. Refer to Appendix B for equation of the linear model.

4.1.2. Clustering Parameter Optimization

The parameters `minClusterSize` and `maxDistance` were optimized by running various combinations of the parameters through the entire pipeline. Pairs of `maxDistance` and `minClusterSize` versus Accuracy are shown in Figure 10. The highlighted point in Figure 10 maximized accuracy with an accuracy value of 96.4% and `minClusterSize` of 10 points and `maxDistance` of 0.8 metres.

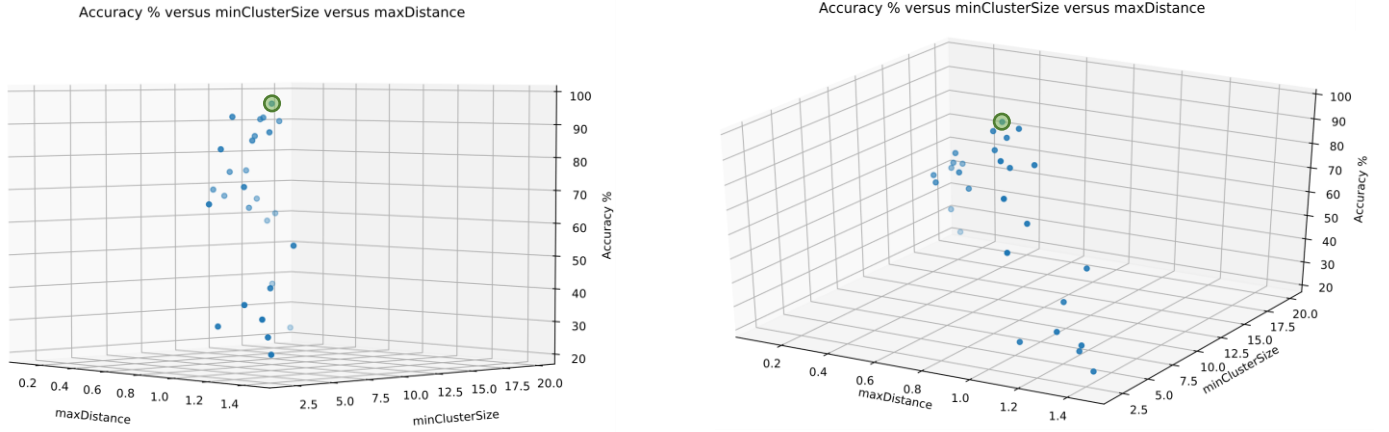


Figure 10: 2 views of the 3D scatterplots showing parameter set that maximized accuracy

4.1.3. Kalman Filter Parameter Optimization

Kalman filter parameters R , the measurement noise covariance, and Q , the system noise covariance were optimized.

The parameters Q and R were optimized in 3 stages:

1. Different combinations of Q and R were run through the pipeline. Then visually, the best performing combinations of Q and R were noted down for further analysis.
2. Location estimates from visually acceptable pairs of Q and R were compared against ground truth values to ensure that our system was producing the accurate location estimates.
3. Similarly performing Q and R combinations were then evaluated by plotting their smoothing characteristic. This curve plots the measured location against the estimated location of a person to assess how much the Kalman filter smooths the values.

The optimal Q and R were:

$$Q = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix} \quad (5)$$

$$R = 5 \quad (6)$$

The plots in Figure 11 show the smoothing characteristic of the designed Kalman filter. Each plot shows the measured versus filtered value in each direction (x and y directions).

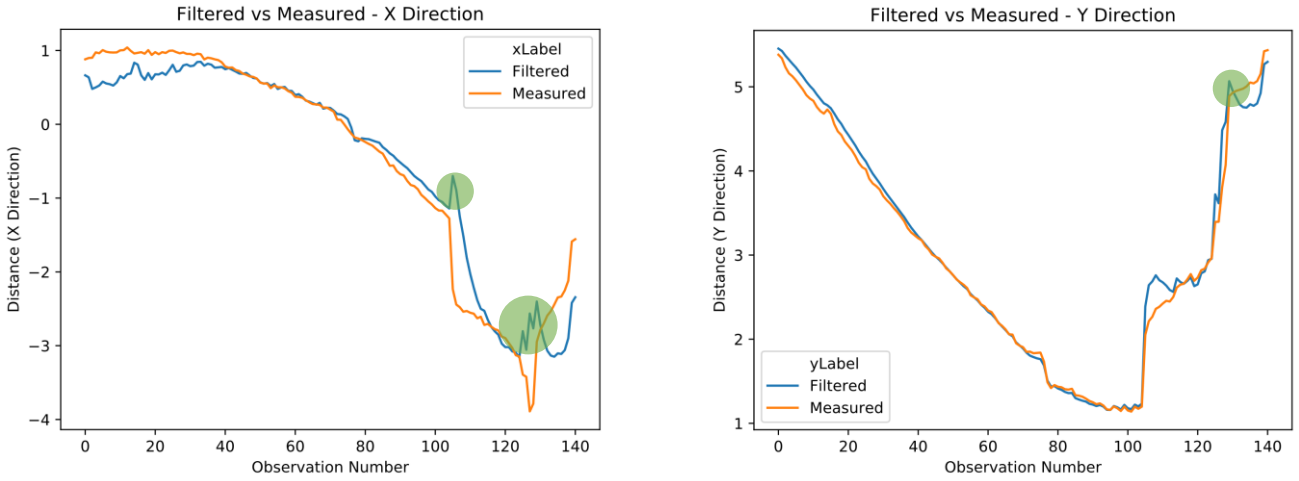


Figure 11: Kalman Filter smoothing characteristic curves

The circled points on the characteristic curves (in Figure 11) are caused by sudden changes in movement. For example, if the occupant went from walking to running or sudden lateral motion changes. To reduce the spikes the Q parameter must be adaptive; varying when the spikes occur to filter them down.

4.2. Embedded Implementation Considerations

The software pipeline had to be implemented on Raspberry Pi 4. Computing considerations were made to fit the processing limitations. Firstly, the entire system was implemented using *NumPy* [24]; A scientific computing package that uses CPython. Many NumPy operations are performed in C. Implementation in C avoids loops, pointer indirection and per-element checking, giving algorithms a substantial speed boost. Secondly, conscious efforts were made to ensure bottlenecks in the pipeline were removed. This was a key reason for the addition of preliminary noise removal. Removal of erroneous clusters improved performance as the clustering algorithm does not need to consider these clusters. Timing analysis using cProfiler [25] was conducted to prove that removing incorrect clusters at an early stage allows for better performance. The timing analysis involved running the software pipeline with and without the preliminary noise removal functionality 1000 times each, over a range of datasets containing between 1 to 5 people. The timing figures were then averaged. Figure 12 shows that the average processing time per frame with preliminary noise removal is significantly shorter than without preliminary noise removal.

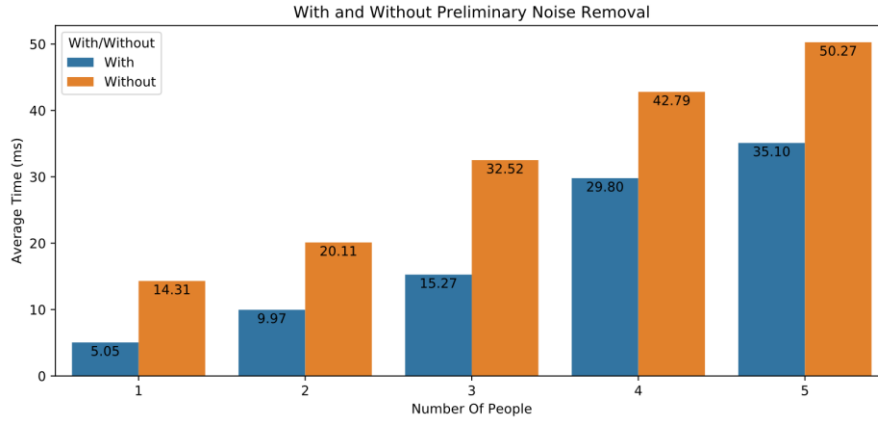


Figure 12: Performance gains with/without primary noise removal

4.3. Testing

Multiple field tests were performed to evaluate the proposed system. The field testing was conducted over three sessions within various rooms around the University of Auckland campus and used numerous volunteers. A variety of situations were tested, including 1 to 5 people walking, 2 to 4 people in a meeting and other combinations of normal indoor behaviour (as shown in Figure 13). A total of 25 datasets consisting of approximately 13,000 frames (samples) were gathered. Numerous datasets of each situation were collected to ensure the reliability of the results.

Artificial environments were constructed (shown in Figure 13) within the field of view of the sensor. Moreover, the sensor was mounted on a tripod around 1.8 meters high with a forward pitch of approximately 10° (shown in Figure 13). Additionally, a camera was also mounted on top of the millimetre-wave sensor to gather ground truth information and document the field-testing process.



Figure 13: Various testing scenarios and experimental setup

4.4. Results

We compared the detection performance of the proposed system against the commercially available Texas Instruments (TI) system using the same millimeter wave sensor, the IWR1642BOOST.

4.4.1. Occupancy Estimation Accuracy

The overall accuracy of the proposed system was 89.3% compared to the 66.7% scored by the TI system. Accuracy figures were calculated by comparing the number of correctly classified samples to the total number of samples collected. Figure 14 below shows the overall experimental accuracy with standard deviation, highlighting that the standard deviation for TI is higher than our system. This means that TI's results are overall less consistent. This can be further observed in the bar-graph in Appendix C. Examine the 2 People Sitting dataset (D1, D2 and D3). Over all 3 sessions (D1, D2 and D3), the experimental accuracy of our system remains relatively constant while the TI system experiences considerable variations in accuracy.

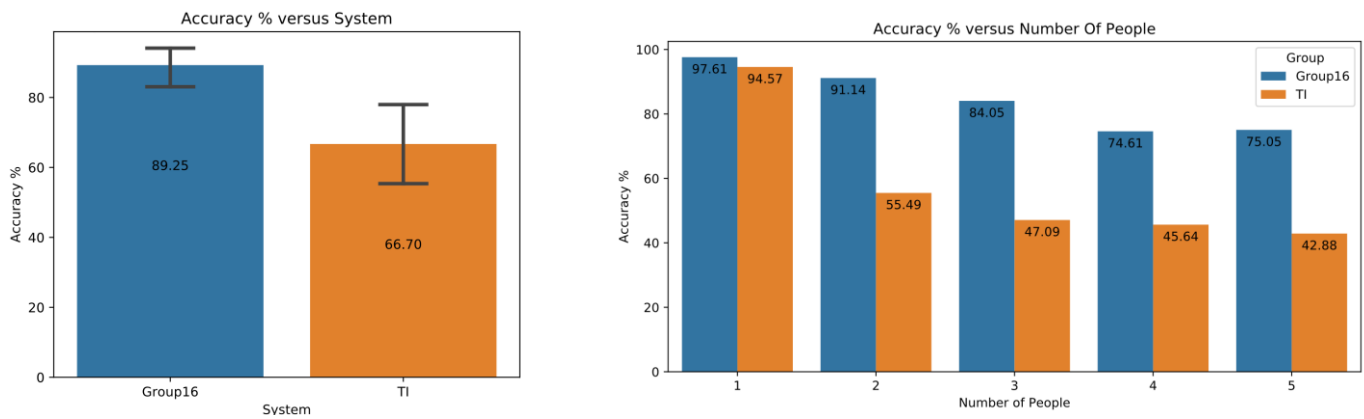


Figure 14: Overall accuracy and Per Group accuracy results

Figure 14 displays that between 1 to 5 people, our system accuracy ranges between 75% to 98%. Outperforming the TI system whose accuracy is between 43% to 95%. However, it also shows that the performance of both systems degrades as the number of people increases. This is because, as number of people increases, a higher proportion of occupants begin occluding one another, leading to a decrease in accuracy. Additionally, the TI system exhibits ghosting when people leave and re-enter the sensor range, decreasing overall accuracy. Ghosting is a phenomenon where the system locates a non-existent person. Ghosting occurs because the TI system has not applied the correct size gating region within its data association method. When a person re-enters, typically they do so outside the given gating region, thus re-entering as a new person, leading to a decrease in accuracy. Our proposed system sets the gating region to a large enough size to where ghosting on re-entry is no longer an issue. However, the issue with setting a sizeable static gating region is the possibility of target ID's swapping on re-entry. This problem was encountered by the proposed system and led

to a decrease in accuracy at higher numbers of people. A better method to fix the issue would be to scale the gating region proportionally, instead of keeping it static.

Moreover, at a higher number of people, we noticed that the amount of missing/corrupted data sent from the millimetre-wave sensor increases. The TI system processes every 4th frame, while our system processes every frame. If a series frames are missing/corrupted, this affects both systems however more so the TI system. Missing/corrupted frames are a significant reason for the discrepancy in accuracy between both systems.

4.4.2. Tracking Accuracy

The average position error of our system is 0.29 metres in the y-direction, and 0.32 metres in the x-direction. In comparison, the average position error of the TI system was 0.3 metres in the y-direction and 0.33 metres in the x-direction. Three datasets (containing 646 samples) were collected from people walking at a location where the position was known. We then ran those datasets through both our system and the TI system calculating the root-mean-squared error in each direction and per site. Location distances from the sensor (centred at the origin) are shown in Table 2.

Table 2: Ground truth for positional testing

Location Number	X (m)	Y (m)
1	-2	3.8
2	1.2	4.2
3	1.5	2.8

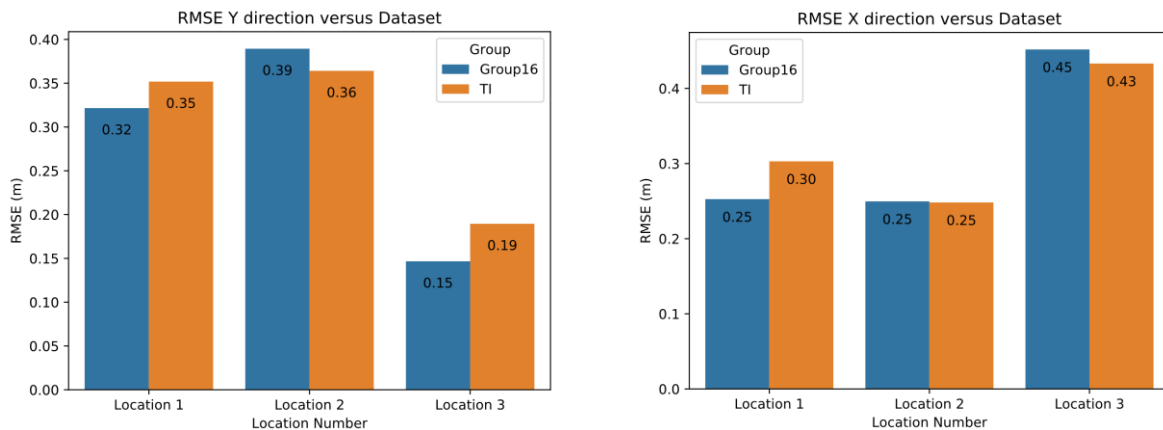


Figure 15: Graphs showing positional error in Y and X direction

Figure 15 shows that in both systems, as a detected person walks vertically further (along the y-axis) from the sensor (Location 3 to Location 1 to Location 2), their error tends to increase.

4.4.3. Timing Analysis

Algorithmic complexity had to be considered because of the embedded aspect of the system. A cycle of the processing stack had to complete in under 50 milliseconds (the frame rate of the millimetre-wave sensor). Any time after 50 milliseconds, the subsequent frame would be missed. Therefore, considerable timing analysis was undertaken to ensure the pipeline worked under 50 milliseconds. Timing analysis was conducted by using the cProfiler [25] functionality in python. Timing data for each frame processed was recorded. Each dataset was run 1000 times to ensure that any fluctuations in the timing data would be minimized. The times were then averaged. Figure 16 shows that runtime tends to increase as the number of people increases. This is expected as an increase in the number of people, increases the number of points to be processed. More importantly, runtime with five people (a high processing load) is below the 50 milliseconds constraint ensuring consecutive frames are not missed.

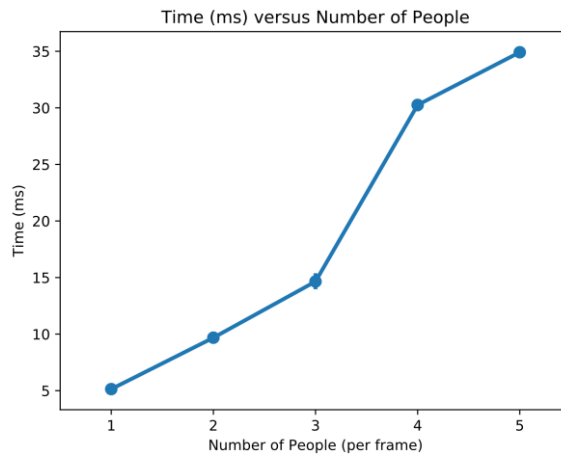


Figure 16: Graph of time per frame (ms) versus number of people

5. Conclusions & Future Directions

In this study, we propose an accurate and portable real-time occupancy detection system for indoor spaces based on millimetre-wave radar. Using the IWR1642BOOST sensor, we first obtain point cloud information. The point cloud is then grouped into various clusters using density-based clustering. These clusters represent

identified occupants. Then we use the recursive Kalman filter and global nearest neighbour data association to track multiple people. Thorough testing showed that the proposed system achieved an overall experimental accuracy of 89.3% with 0.32 metres and 0.29 metres of x-y positioning error, respectively.

Millimetre-wave radar technology is relatively new, and therefore, more research is required to create a commercial-grade system for occupancy detection. Interesting future work consists of data fusion from multiple millimetre-wave sensors and applying deep learning for tracking. Multiple millimetre-wave radar sensors would decrease the effect of occlusions at higher numbers of people and increase the effective field of view. Additionally, a deep learning approach could use convolutional neural networks for tracking.

Acknowledgements

I would like to acknowledge my partner, Abin Thomas for the great work and support he provided. I would also like extend thanks to Mr. Terry Huang and Professor Sing Kiong Nguang. The advice given to us was invaluable. I look forward to working with them next year.

References

- [1] E. Yavari, C. Song, V. Lubecke, and O. Boric-Lubecke, "Is There Anybody in There?: Intelligent Radar Occupancy Sensors," *IEEE Microw. Mag.*, vol. 15, no. 2, pp. 57–64, 2014.
- [2] A. Santra, R. Vagarappan Ulaganathan, and T. Finke, *Short-Range Millimetric-Wave Radar System for Occupancy Sensing Application*, vol. PP. 2018.
- [3] T. Yokoishi, J. Mitsugi, O. Nakamura, and J. Murai, "Room occupancy determination with particle filtering of networked pyroelectric infrared (PIR) sensor data," in *SENSORS, 2012 IEEE*, 2012, pp. 1–4.
- [4] O. Shih and A. Rowe, "Occupancy Estimation Using Ultrasonic Chirps," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, 2015, pp. 149–158.
- [5] R. Valle, "ABROA: Audio-based room-occupancy analysis using Gaussian mixtures and Hidden Markov models," in *2016 Future Technologies Conference (FTC)*, 2016, pp. 1270–1273.
- [6] S. H. Ryu and H. J. Moon, "Development of an occupancy prediction model using indoor environmental data based on machine learning techniques," *Build. Environ.*, vol. 107, pp. 1–9, 2016.
- [7] M. K. Masood, C. Jiang, and Y. C. Soh, "A novel feature selection framework with Hybrid Feature-Scaled Extreme Learning Machine (HFS-ELM) for indoor occupancy estimation," *Energy Build.*, vol. 158, pp. 1139–1151, 2018.
- [8] Y. Benezeth, H. Laurent, B. Emile, and C. Rosenberger, "Towards a sensor for detecting human presence and characterizing activity," *Energy Build.*, vol. 43, no. 2–3, pp. 305–314, 2011.
- [9] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–7.
- [10] D. Caicedo and A. Pandharipande, "Ultrasonic array sensor for indoor presence detection," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 175–179.
- [11] Z. Han, R. X. Gao, and Z. Fan, "Occupancy and indoor environment quality sensing for smart buildings," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2012, pp. 882–887.

- [12] K. Weekly, H. Zou, L. Xie, Q.-S. Jia, and A. M. Bayen, "Indoor occupant positioning system using active RFID deployment and particle filters," in *2014 IEEE International Conference on Distributed Computing in Sensor Systems*, 2014, pp. 35–42.
- [13] N. Li, G. Calis, and B. Becerik-Gerber, "Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations," *Autom. Constr.*, vol. 24, pp. 89–99, 2012.
- [14] Z. Zhen, Q. Jia, C. Song, and X. Guan, "An Indoor Localization Algorithm for Lighting Control using RFID," in *2008 IEEE Energy 2030 Conference*, 2008, pp. 1–6.
- [15] R. Tomastik, S. Narayanan, A. Banaszuk, and S. Meyn, "Model-based real-time estimation of building occupancy during emergency egress," in *Pedestrian and Evacuation Dynamics 2008*, Springer, 2010, pp. 215–224.
- [16] T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan, "Real-time building occupancy sensing using neural-network based sensor network," in *2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2013, pp. 114–119.
- [17] K. P. Lam et al., "Information-theoretic environmental features selection for occupancy detection in open offices," in *Eleventh International IBPSA Conference*, 2009, pp. 27–30.
- [18] E. Yavari, H. Jou, V. Lubecke, and O. Boric-Lubecke, "Doppler radar sensor for occupancy monitoring," in *2013 IEEE Topical Conference on Power Amplifiers for Wireless and Radio Applications*, 2013, pp. 145–147.
- [19] P. Zhao et al., "mID: Tracking and Identifying People with Millimeter Wave Radar," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 33–40.
- [20] Texas Instruments, "People Tracking and Counting Reference Design Using mmWave Radar Sensor." 2018.
- [21] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering.," *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.
- [22] P. Liu, D. Zhou, and N. Wu, "VDBSCAN: varied density based spatial clustering of applications with noise," in *2007 International conference on service systems and service management*, 2007, pp. 1–4.
- [23] L. Campagnola, "PyQtGraph-scientific graphics and GUI library for python." 2016.
- [24] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation," *Comput. Sci. Eng.*, vol. 13, no. 2, p. 22, 2011.
- [25] M. Wagner, G. Llort, E. Mercadal, J. Giménez, and J. Labarta, "Performance Analysis of Parallel Python Applications," *Procedia Comput. Sci.*, vol. 108, pp. 2171–2179, 2017.

Appendix A: Kalman Filter Equations

This appendix contains material relating to the recursive Kalman Filter.

Predict

$$x_{pred} = Ax \tag{7}$$

$$P_{pred} = APA^T + Q \tag{8}$$

$$v = z - Cx_{pred} \tag{9}$$

$$S = R + CP_{pred}C^T \tag{10}$$

Update

$$K = P_{pred}C^TS^{-1} \tag{11}$$

$$x_{new} = x_{pred} + Kv \tag{12}$$

$$P_{new} = P_{pred} - KSK^T \tag{13}$$

Appendix B: SNR Filtering Model

This appendix contains material relating to the equation and model assumptions for the SNR filtering model.

The equation of the model is shown below:

$$\log(SNR_i) = \beta_0 + \beta_1 \times Range_i + \varepsilon_i \quad (14)$$

Where $\varepsilon_i \sim iid N(0, \sigma^2)$

Appendix C: Accuracy Graphs

This appendix contains the accuracy bar graph over all datasets for our system versus the Texas Instruments system.

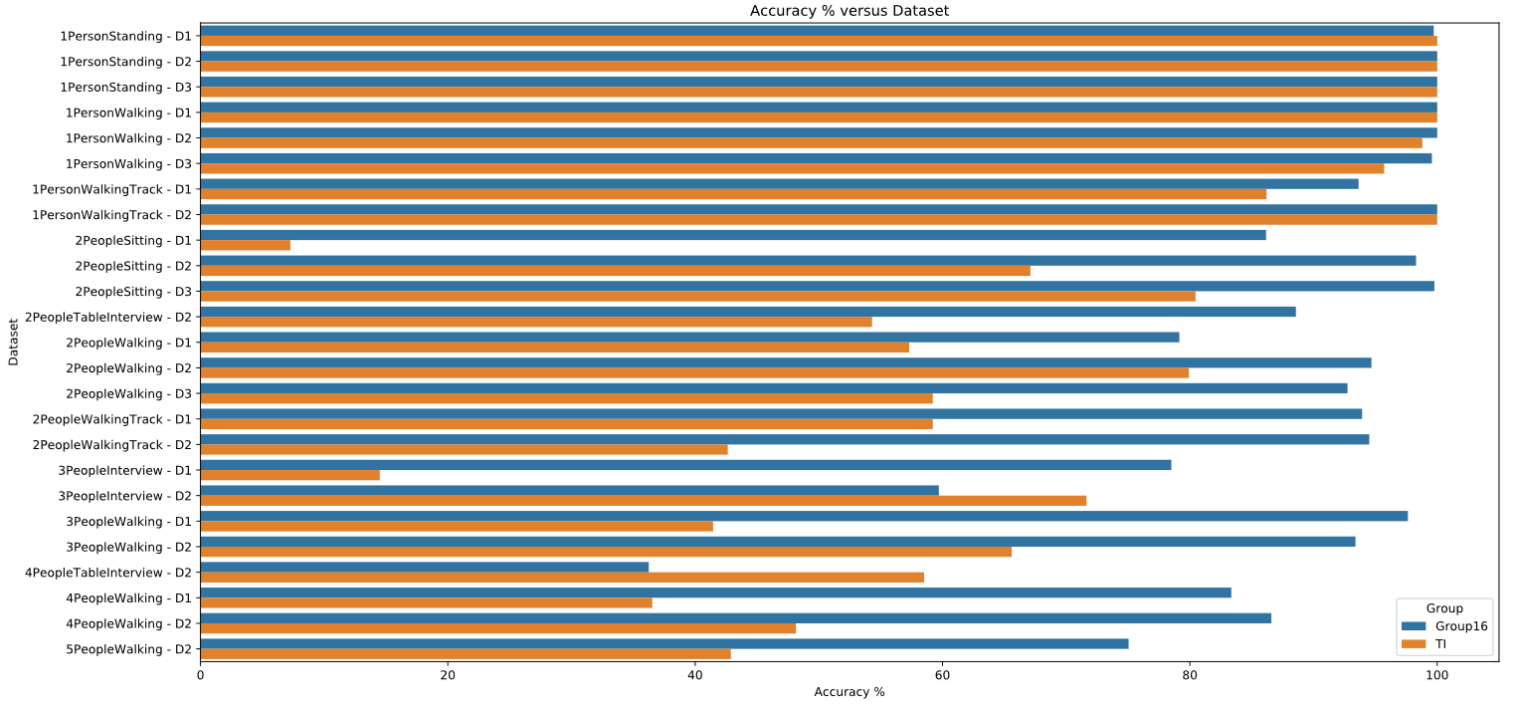


Figure 17: Accuracy graph over all datasets for our system versus TI