

Research Article

Indoor Detection and Tracking of People Using mmWave Sensor

Xu Huang ^{1,2}, Hasnain Cheena,¹ Abin Thomas,¹ and Joseph K. P. Tsoi ¹

¹Department of Electrical and Computer Engineering, The University of Auckland, Auckland 1010, New Zealand

²Faulty of Information Engineering, University of Shandong Ying Cai, Shandong, Jinan 250104, China

Correspondence should be addressed to Joseph K. P. Tsoi; ktso005@aucklanduni.ac.nz

Received 9 December 2020; Revised 3 January 2021; Accepted 12 January 2021; Published 3 February 2021

Academic Editor: Bin Gao

Copyright © 2021 Xu Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new indoor people detection and tracking system using a millimeter-wave (mmWave) radar sensor. Firstly, a systematic approach for people detection and tracking is presented—a static clutter removal algorithm used for removing mmWave radar data's static points. Two efficient clustering algorithms are used to cluster and identify people in a scene. The recursive Kalman filter tracking algorithm with data association is used to track multiple people simultaneously. Secondly, a fast indoor people detection and tracking system is designed based on our proposed algorithms. The method is lightweight enough for scalability and portability, and we can execute it in real time on a Raspberry Pi 4. Finally, the proposed method is validated by comparing it with the Texas Instruments (TI) system. The proposed system's experimental accuracy ranged from 98% (calculated by misclassification errors) for one person to 65% for five people. The average position errors at positions 1, 2, and 3 are 0.2992 meters, 0.3271 meters, and 0.3171 meters, respectively. In comparison, the Texas Instruments system had an experimental accuracy ranging from 96% for one person to 45% for five people. The average position errors at positions 1, 2, and 3 are 0.3283 meters, 0.3116 meters, and 0.3343 meters, respectively. The proposed method's advantage is demonstrated in terms of tracking accuracy, computation time, and scalability.

1. Introduction

Indoor detection and tracking of people are useful solutions to energy assignment, health, and safety [1]. Studies show that an indoor detection and tracking system can reduce energy usage for lighting and Heating, Ventilation, and Air Conditioning (HVAC) systems by more than 30% [2]. Additionally, these systems can also improve security applications by giving emergency systems the ability to make more well-informed decisions. So that can enhance the response of emergency systems by providing them with real-time location information of people, where they are going, and the densities of people at different sites to decide whether they are safe or not. Moreover, indoor detection and tracking systems could also help health care businesses monitoring the elderly when they fall. For example, based on location information, nursing staff could make a decision ensuring their safety.

Researchers have studied various types of sensing technologies for indoor object detection measurements, such as passive infrared (PIR) [3], optical cameras [4, 5], LIDAR

[6], Wi-Fi [7], and 10 GHz-to-24 GHz microwave [8]. However, all these technologies have challenges with inaccuracy, privacy, environmental robustness, and system complexity [9]. For instance, the HD camera system and other technologies such as Wi-Fi, Bluetooth, and UWB are used for positioning [10–12]. In these studies, Machine Learning (ML) methods are employed to detect people. These methods include decision trees, hidden Markov models, and convolutional neural networks. Machine learning is a computationally intensive process and cannot readily be implemented onto an embedded system.

Moreover, typically camera-based tracking systems require a clear view and the right lighting conditions displayed in [13], where the system uses background subtraction and the Lucas and Kanade tracking algorithm to determine indoor human counting. 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,

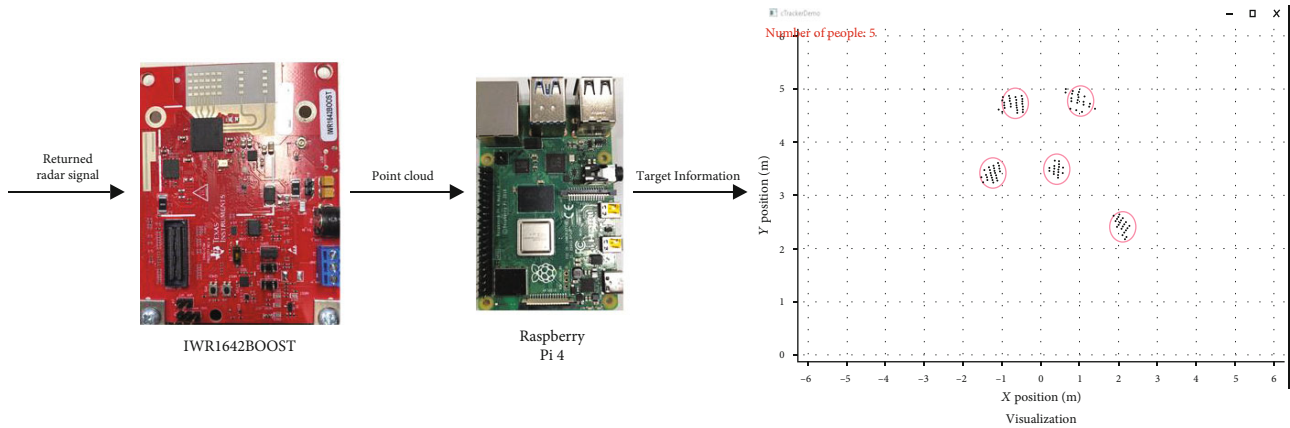


FIGURE 1: Flow of hardware information.

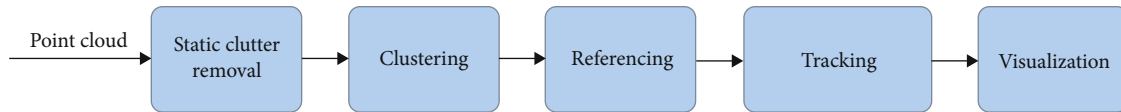


FIGURE 2: Framework of the data process.

research is being done to discover whether it is possible to use lower resolution cameras to circumvent privacy issues [14].

Motivated by this, this research chose the onboard millimeter-wave (mmWave) radar sensor (IWR1642-BOOST) [15] as the sensing technology [16]. mmWave is a remote wireless sensing technology that has raised lots of attention from academia and industry due to its exceptional advantages. Compared to the existing wireless sensing technologies, this particular radar technology can overcome environmental occlusion problems. We aim to explore fast and robust people detection and tracking models, algorithms, and application guidance using mmWave sensors for indoor applications.

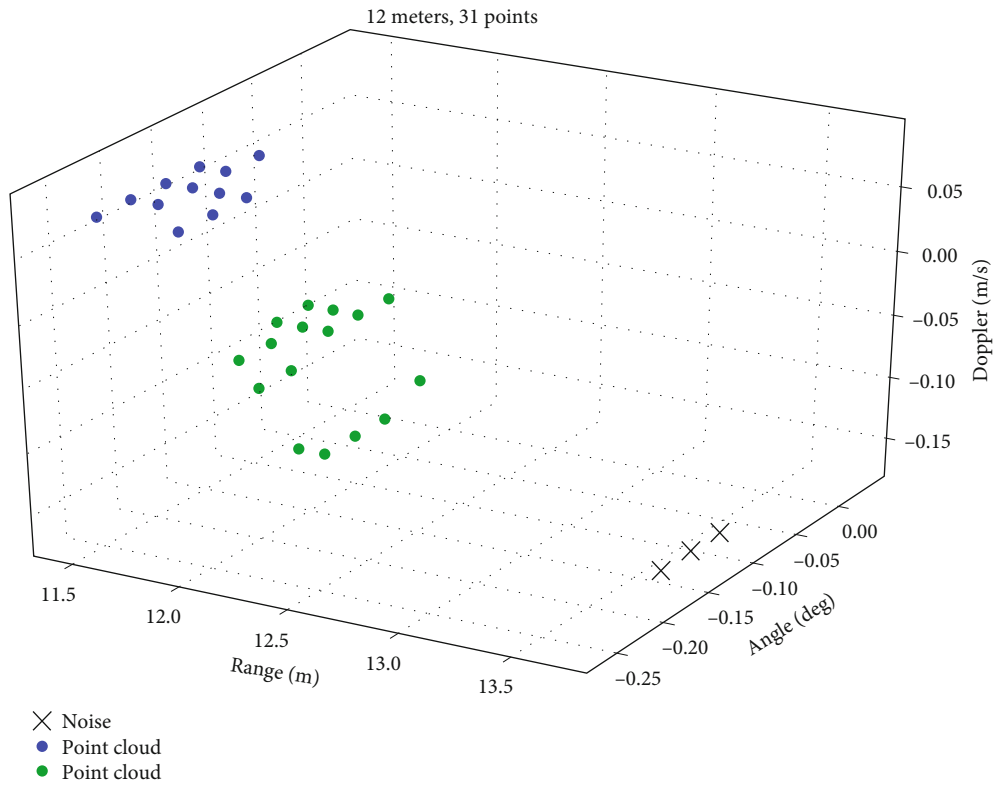
Ongoing research in object detection and tracking data process technologies are mainly focused on vision-based methods [4]. There are currently only limited studies using mmWave radar data for indoor detection and tracking of people. In [17], Wei and Zhang set up a new high-precision passive tracking method (mTrack) and used highly directional 60 GHz millimeter-wave radars to run a discrete beam scanning mechanism to pinpoint the object's initial location and track its trajectory. However, it is based on a signal-phase model. Hence, it is not suitable for applying detection and tracking indoors. In [18], Palacios et al. developed two indoor localization algorithms tailored to mm-wave propagation characteristics based on commercial 60 GHz mm-wave hardware. However, its experiment results only considered location error, and the system computation load is not mentioned. The most related work is people counting and tracking using a mmWave radar sensor by Texas Instruments (TI) [19]. The system employs density-based clustering (DBSCAN) with an extended Kalman filter (EKF), and it reported an accuracy of 51% to 99% between 1 and 5 people. However, its accuracy is questionable since only DBSCAN [20] is used to clustering the varying density data.

Moreover, its portability and scalability are limited due to the use of EKF to convert the polar measurement to Cartesian coordinates. The conversion is taken for ease of use, yet it brings additional computation load and process noise.

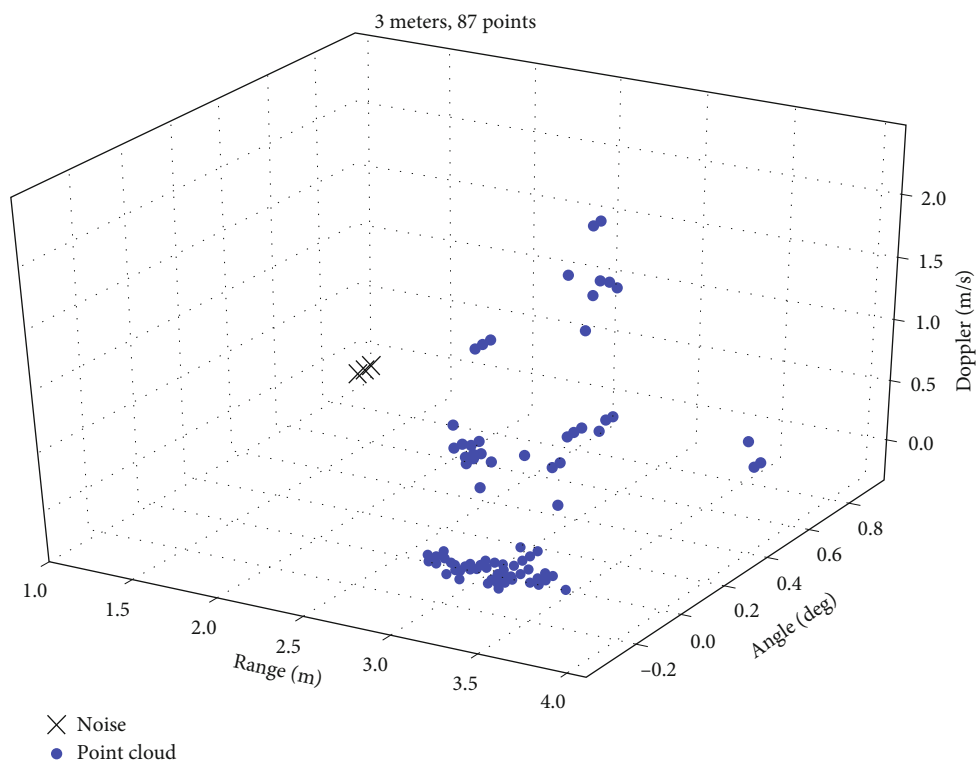
This paper includes two main contributions. Firstly, we present a systematic approach to detecting and tracking people indoors using a mmWave radar sensor. Two efficient clustering algorithms are proposed here to provide high accuracy and shallow processing time; the recursive Kalman filter (RKF) tracking algorithm performs much better than the EKF in algorithmic complexity and computation time. Moreover, a fast indoor people detection and tracking system was designed based on our proposed algorithms. Furthermore, the system can operate on an embedded platform, the Raspberry Pi 4, creating computing constraints to introduce a portability and scalability aspect. Comparing the results to the commercially available system from TI shows that the method is faster, more accurate, and less heavy than the TI system.

2. Methodology

2.1. Hardware Framework. The hardware system consists of the millimeter-wave radar sensor (IWR1642BOOST), a Raspberry Pi 4 (1.5GHz, 4GB RAM), and a monitor. The flow of hardware 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.



(a)



(b)

FIGURE 3: Comparison of the numbers of clustered points at different distances: (a) data points at 12 meters from mmWave sensor; (b) data points at 3 meters from mmWave sensor.

Parameters:

- maxDistance- the largest searching distance
- minClusterSize- the minimum number of points to classify as a cluster

Steps:

- (1) Set the maxDistance and minClusterSize parameters for the clustering algorithm.
- (2) Randomly select a point c that has not been marked a cluster or been designated as an outlier (noise).
- (3) Compute its neighborhood to determine if it is a core point. If yes, start a cluster around this point.
If no, mark the point as an outlier.
- (4) If p is a core point, a cluster is formed, expand the cluster by adding all directly reachable points to the cluster.
- (5) If an outlier is added, change that point's status from outlier to border point.
- (6) Repeat steps 2-5 until all points are either assigned to a cluster or designated as an outlier.
- (7) Calculate the mean of each cluster.

ALGORITHM 1: Clustering: DBmeans.

Parameters:

- maxDistance- the largest searching distance
- minClusterSize- the minimum number of points to classify as a cluster

Steps:

- (1) Set the maxDistance and minClusterSize parameters for the clustering algorithm.
- (2) Randomly select a point p that has not been marked a cluster or been designated as an outlier (noise).
- (3) Compute its neighborhood to determine if it is a core point. If yes, start a cluster around this point.
If no, mark the point as an outlier.
- (4) If p is a core point, a cluster is formed, expand the cluster by adding all directly reachable points to the cluster.
- (5) If an outlier is added, change that point's status from outlier to border point.
- (6) Repeat steps 2-5 until all points are either assigned to a cluster or designated as an outlier.
- (7) Randomly select a point c in each cluster as a medoid.
- (8) Assign each of the remaining points (nonmedoid) in every cluster represented by the nearest medoid.
- (9) Randomly select a nonmedoid point O_{random} in every cluster.
- (10) Consider each of the current medoids O_j in every cluster: compute the total cost S of swapping O_j with O_{random} , includes the cost contributions of reassigning nonmedoid points caused by the swap.
If $S < 0$, then swap O_j with O_{random} to form the new medoid.
- (11) Repeat steps 8-10, until no change.

ALGORITHM 2: Clustering: DBmedoids.

2.2. *System Data Process Framework.* The flow chart in Figure 2 depicts the systematic approach used in this paper to process and analyze mmWave sensor raw data (point cloud), then the sensor transmits the point cloud data to the Raspberry Pi. This paper mainly focuses on the clustering and referencing algorithms, tracking algorithm, and timing analysis of the merged approach for real time and portability, then compares it to the TI method. Firstly, the point cloud information from the mmWave sensor is parsed and then processed for static clutter removal. The points are then grouped in the clustering+referencing module, and then finally, the people's points are tracked in the tracking module, from which the people number is derived.

2.3. *Static Clutter Removal.* The static clutter removal model is aimed at excluding as many as static points as possible from the background. It requires range information since it filters out nonrange changing (static) objects from the scene. The steps of the static clutter removal algorithm are listed as follows.

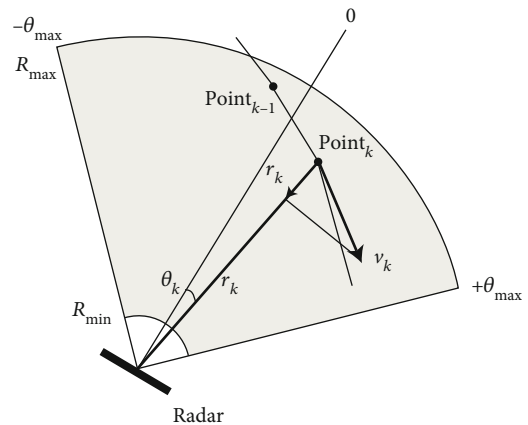


FIGURE 4: Radar geometry in 2D.

Step 1. Range processing performs Fast Fourier Transform (FFT) on Analog to Digital Converter (ADC) samples per antenna per chirp. FFT output is a set of range bins.

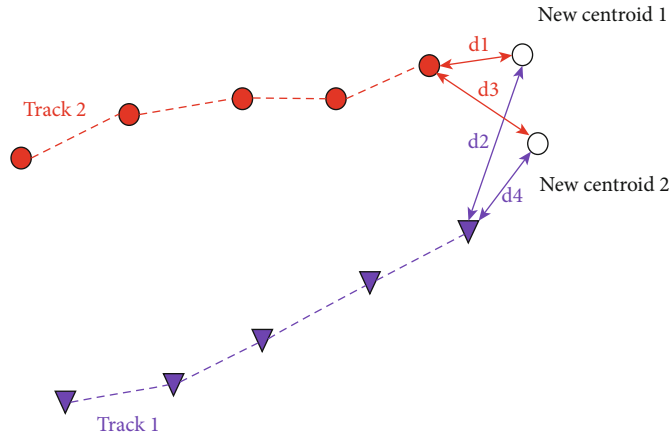


FIGURE 5: Simplified GNN diagram.

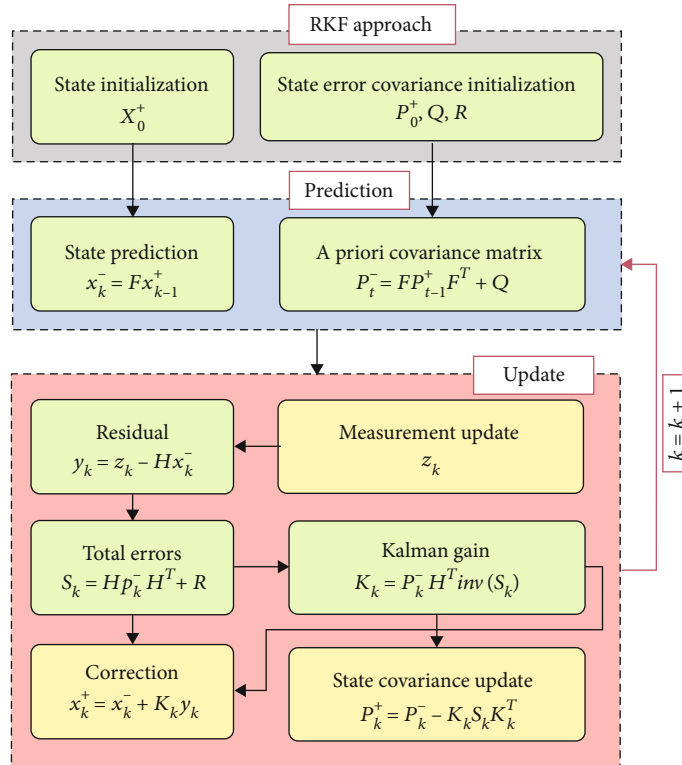


FIGURE 6: Implementation flowchart of the proposed RKF.

Step 2. Perform static clutter removal by subtracting the estimated Direct Current (DC) component from each range bin.

Step 3. Range processing results in local scratch buffers are Enhanced Direct Memory Access (EDMA) to the radar data cube with transpose.

2.4. Clustering and Referencing. The clustering stage is aimed at identifying the number of people in a scene, and since a single centroid is needed to track each person, a referencing process is required.

2.4.1. Clustering. Due to the mmWave sensor field of view, the data's density varies from time and distance against the sensor. For example, the closer the people are to the sensor, the more dense points can be collected. On the other hand, as distance increases, only a few points can be obtained, especially for smaller objects. To demonstrate, Figure 3 shows the different total number of collected points (cluster density) of people located at different distances from the sensor. There are only 31 points (blue and green) of two people around 12m away from the sensor, while another person is only 3m away from the sensor, which collected 87 points (blue and green). It showed that a denser cluster represents a person closer to the sensor. In contrast, a person

Steps:

- (1) Calculate the distances between each old centroid (objects) and new centroid (observations) in popular coordinate system (e.g., d1, d2, d3, and d4). $d^2 = r_1^2 + r_2^2 - 2r_1r_2 \cos(\theta_1 - \theta_2)$.
- (2) Find the one with the global smallest distance from the total distances (e.g., d1).
- (3) Associate the object with the new centroid linked by this distance (e.g., associate Track 2 and new centroid 1).
- (4) Repeat steps 2-3 until all unassociated new centroids and objects are associated (e.g., associate Track 1 and new centroid 2).

ALGORITHM 3: Data association: simplified GNN.

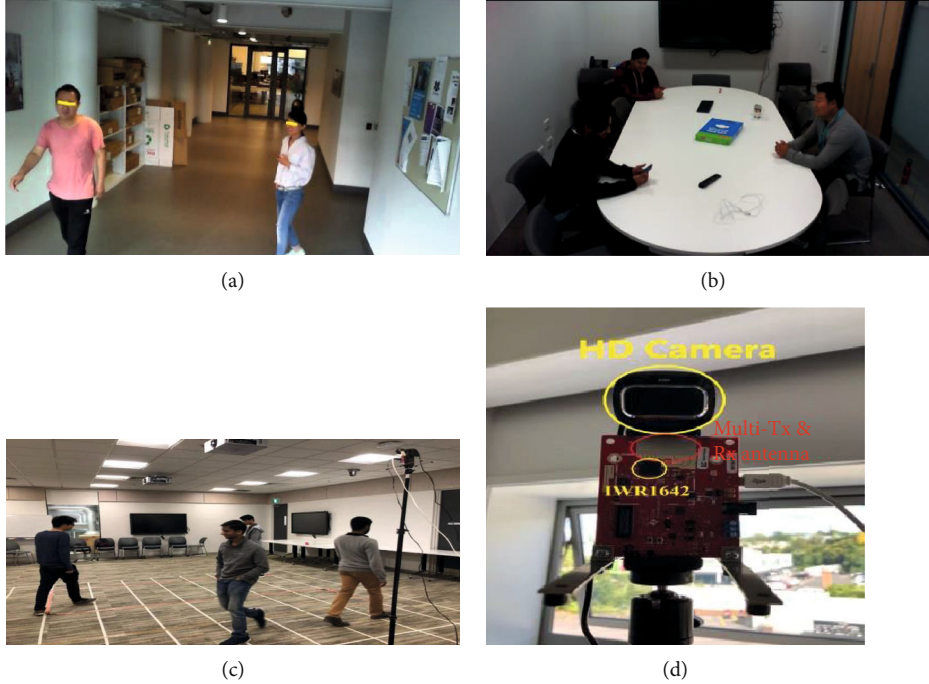


FIGURE 7: Various data collection sites and experimental setup: (a) corridor; (b) meeting room; (c) seminar room; (d) experimental setup.

who is further away is represented by a less dense and more variable cluster.

The black cross stands for the density-based noise points after the clustering stage. These clusters are identified to be the noise of points that are too small to represent people. The density-based noise identifies works by treating each point as a node and then calculating the distance matrix between itself and all the other nodes. The distance between each node is the difference in displacement in the x -direction and the displacement in the y -direction. If a node is within a distance threshold of 0.2 m to the other nodes, those nodes are extracted.

2.4.2. Referencing. After clustering, all detected people are represented by clusters. A reference point on the X and Y plane needs to be found to locate each cluster's position. This reference point will later be used for tracking clusters and extracting trajectory information. The reference point can be the mean center of a cluster and also can be the real center point (both can be called centroid) of a cluster. For people clusters, both can be used as the reference point.

The two density-based clustering and referencing algorithms that we implemented are DBmeans and DBmedoids, respectively. The algorithms can get the centroid location or the centroid point of a cluster with a shallow misclassification rate. The two algorithms are presented as Algorithm 1 and Algorithm 2.

2.5. Tracking

2.5.1. Recursive Kalman Filter. The tracking stage is necessary to locate people as they move through the indoor space and maintain accurate and reliable measurements. In this paper, a recursive estimation method with Kalman filter (RKF) plus a motion model is applied for motion state prediction and estimation of people. Since there are inconsistencies in the rate at which data was lost from the sensor, we decided to recursively calculate the error covariance matrix and Kalman gain in each update stage. We could get a more accurate update compared to a static Kalman gain.

We opted for the RKF, as it has not been researched in mmWave indoor people tracking, and we theorized that we could make it lightweight for real-time application. The

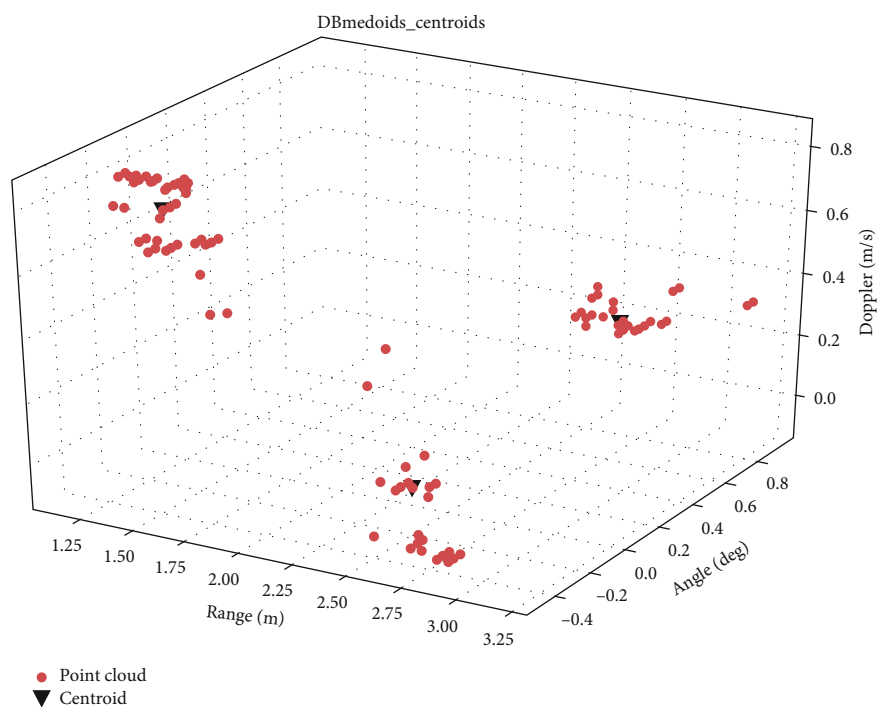
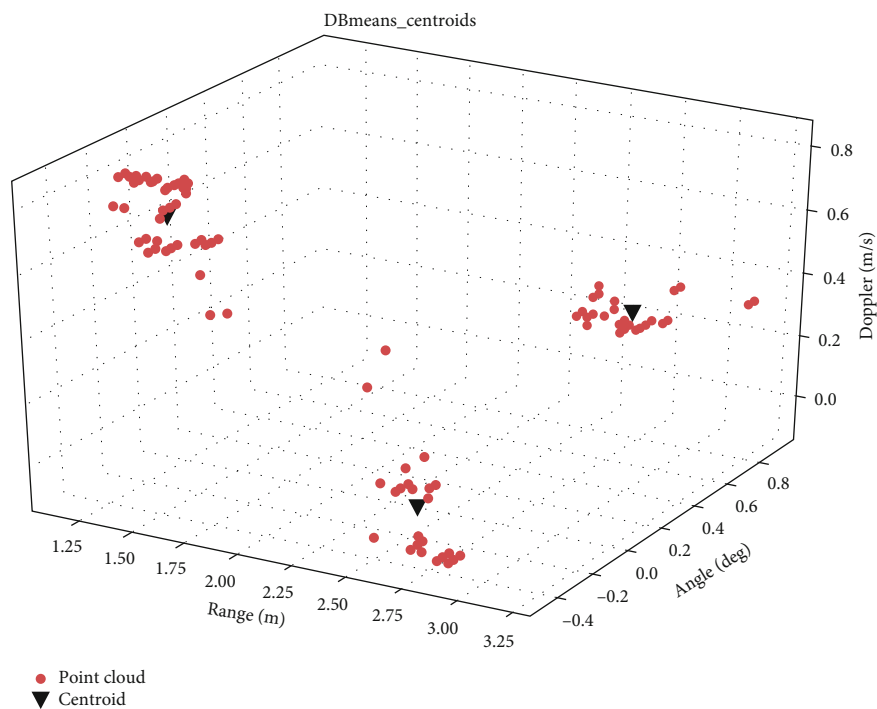
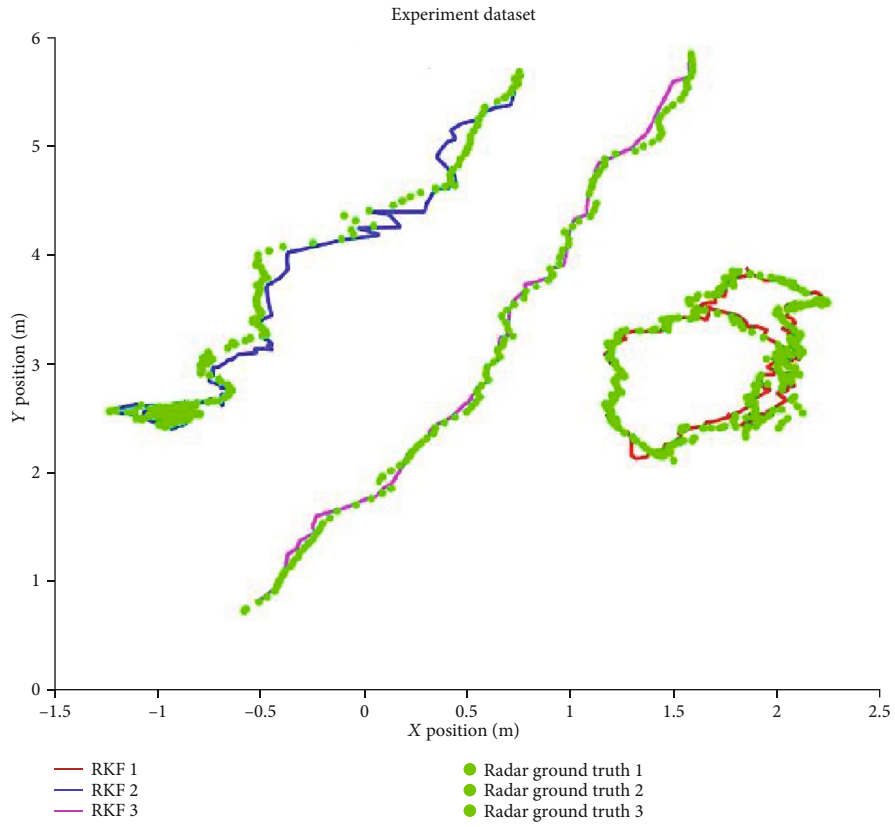


FIGURE 8: Example frame of the DBmeans and DBmedoids algorithms: (a) DBmeans; (b) DBmedoids.

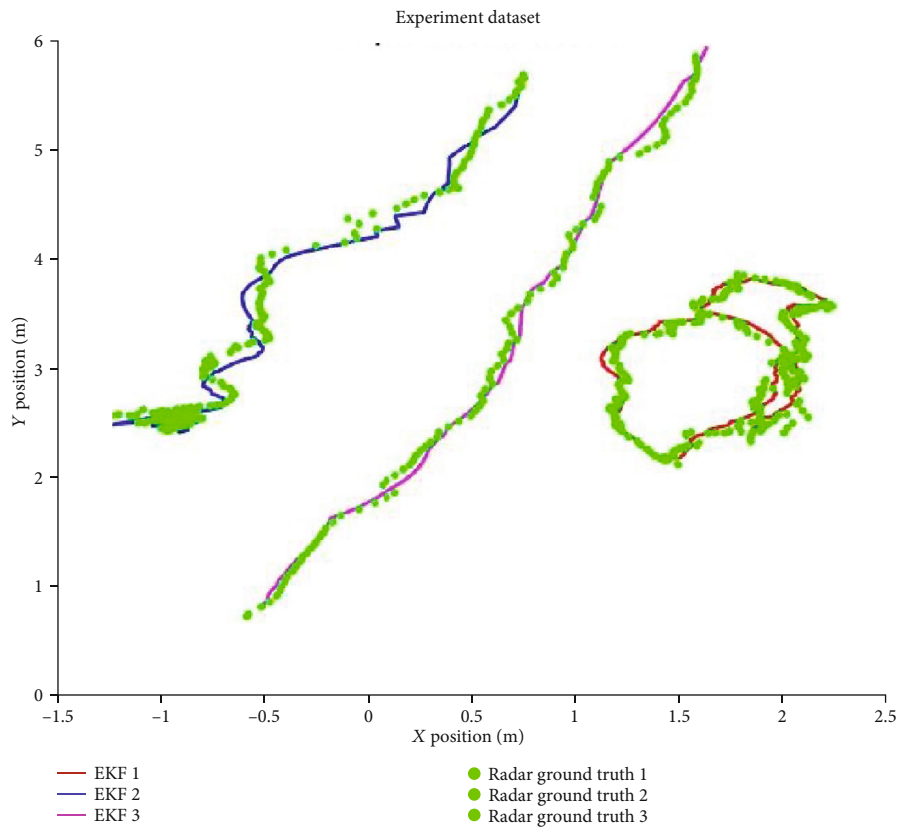
RKF is well-suited for indoor people tracking when using a constant velocity (CV) model, and we also considered an acceleration model by random noise. Moreover, it can improve accuracy by avoiding the EKF's process errors caused by linearization by keeping computation in the polar coordinates, illustrated in Figure 4. Figure 4 illustrates the single reflection point at time k . Multiple reflection points

TABLE 1: Comparison between the DBmeans and DBmedoids algorithms.

Algorithms	Total frames	Misclustering	Time (ms)	Accuracy
DBmeans	341	52	29.11	84.75%
DBmedoids	341	59	121.67	82.70%



(a)



(b)

FIGURE 9: RKF vs. EKF: (a) RKF; (b) EKF.

represent real-life radar objects. Each point is represented by range r ($R_{\min} < r < R_{\max}$), angle ($-\theta_{\min} < \theta < \theta_{\max}$), and radial velocity \dot{r} (range rate). To employ RKF, we keep the raw data processing from detection to tracking under the polar system and keep the visualization under the Cartesian coordinates for the best view.

The system state in the polar system at step k can represent as

$$x_k = [r \dot{r} \theta \dot{\theta}]^T. \quad (1)$$

The motion state model and observation model of people can be built as follows:

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + Q, \\ y_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_k + R, \end{aligned} \quad (2)$$

where Δt is the mmWave sensor sampling time interval and was set to 50 ms. Q and R are the system noise covariance matrix and measurement noise covariance matrix, respectively.

F is a transition matrix,

$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where H is a measurement matrix,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (4)$$

An implementation flowchart of the proposed RKF algorithm is summarized in Figure 5. As shown, the update step involves recursively calculating the Kalman gain K , then calculating the current data frame's state x_k^+ and the error covariance matrix P_k^+ . Recalculating the Kalman gain and error covariance can give the estimate system more robust and practical flexibility. Moreover, if no measured data are available, the estimated values are used as the updated values. The algorithm is described as follows.

In the initialization step, the mean values x_0^+ and covariance matrix P_0^+ of the states are set up at $k=0$, where the superscript “+” indicates that the estimate is a posteriori, and P is the error state covariance matrix.

TABLE 2: Comparison between the RKF and the EKF.

Algorithms	Total frames	Number of people	RMSE	Total time (ms)
RKF	245	3	0.0471	62.5
EKF	245	3	0.0488	281.3

In the prediction step, the state and its covariance matrix at $k-1$ (x_{k-1}^+ , P_{k-1}^+) are projected one step forward to obtain the a priori estimates at k (x_k^- , P_k^-).

In the update step, the actual measurement is compared with predicted measurement based on the a priori estimate. The difference is used to obtain an improved a posteriori estimate as in Figure 6. Symbols z_k and S_k are the measurement vector and innovation covariance, respectively.

2.5.2. Data Association. Since there could be multiple people at any time, and the Kalman filter can only track a single person at a time; therefore, we implement a lightweight data association approach with a recursive Kalman filter to work on multiple objects. The global nearest neighbor (GNN) data association algorithm used in our system is a simplified version and based on the centroid data after the clustering and referencing step. The simplified GNN diagram is shown in Figure 5, and the algorithm description is shown in Algorithm 3.

After GNN is processed, the associated centroids can be passed through the update step of the RKF to be a multiobject tracker. Each track goes through a life cycle of events. At the maintenance step, we decide to change the state or delete the track that is not used anymore.

3. Experiment and Evaluation

3.1. Experiment Setup. To evaluate our algorithms' performance, we set up experiments at three different data collection sites around the University of Auckland Newmarket campus to model various real indoor scenarios. The mmWave sensor data was captured using the TI IWR1642-BOOST. The IWR1642BOOST radar sensor includes an FMCW transceiver, operating at 76 GHz to 81 GHz (4 GHz available bandwidth) with four receive channels and two transmit channels. It outputs a data frame containing the point cloud, with information for each detected point, including range, azimuth angle, and Doppler velocity. Various settings and modes, such as different ranges, can be selected for using the chirp configuration parameters. There are settings for short-range (10 m), midrange (30 m), and long-range (80 m), albeit at the expense of a narrower field of view. For indoor detection and tracking, we opted for a range of 6 m to maximize the resolution and view field.

For each of the data collection sites, the mmWave sensor was mounted on a tripod and elevated to a height of 1.8 to 2 m. The sensor is placed in the environment so that the field of view covers the range r of 1 to 6 meters and an azimuth θ of -60° to 60° , oriented towards the direction in which people would enter the scene. Additionally, an HD camera was also mounted on top of the millimeter-wave sensor to gather



(a)



(b)



(c)

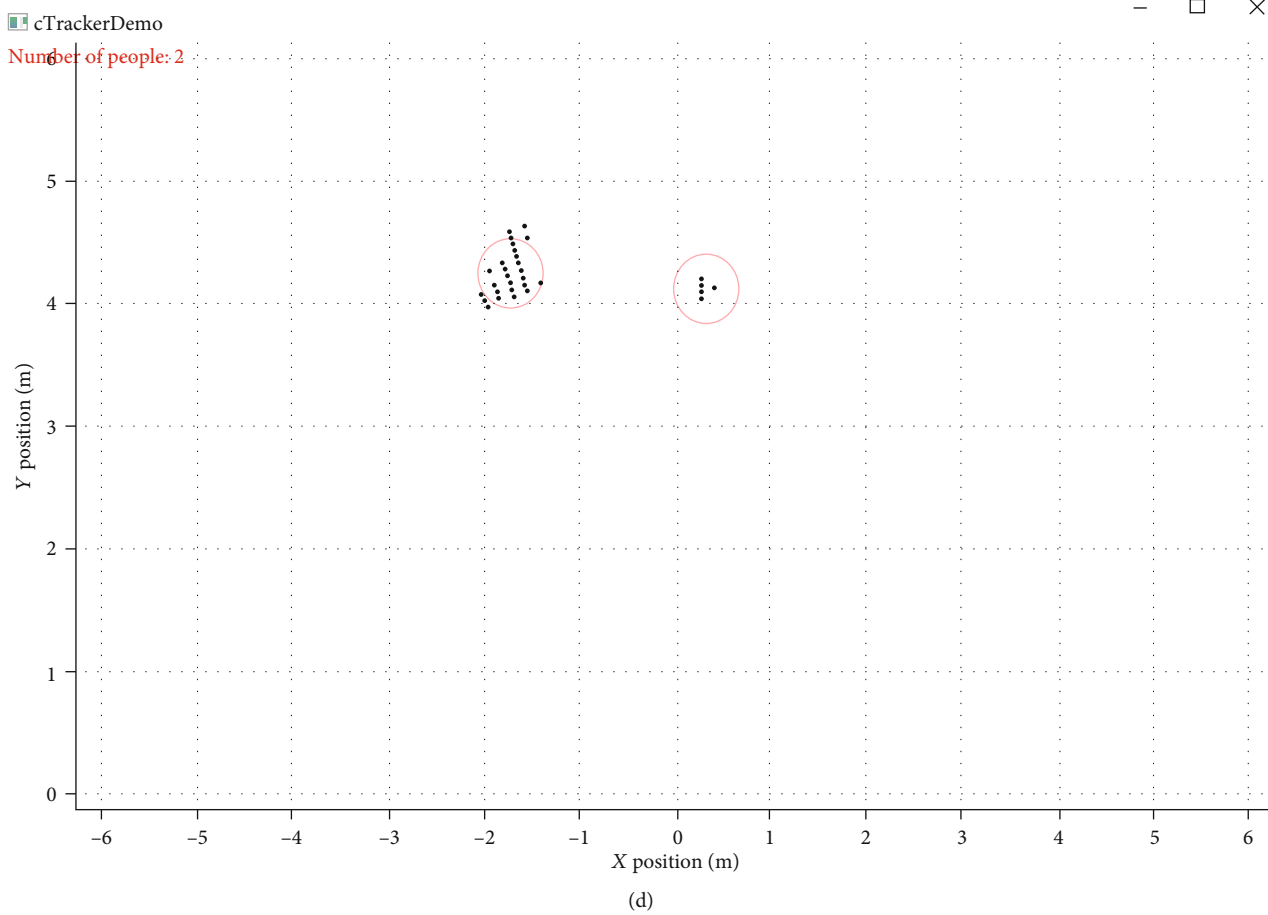


FIGURE 10: Continued.

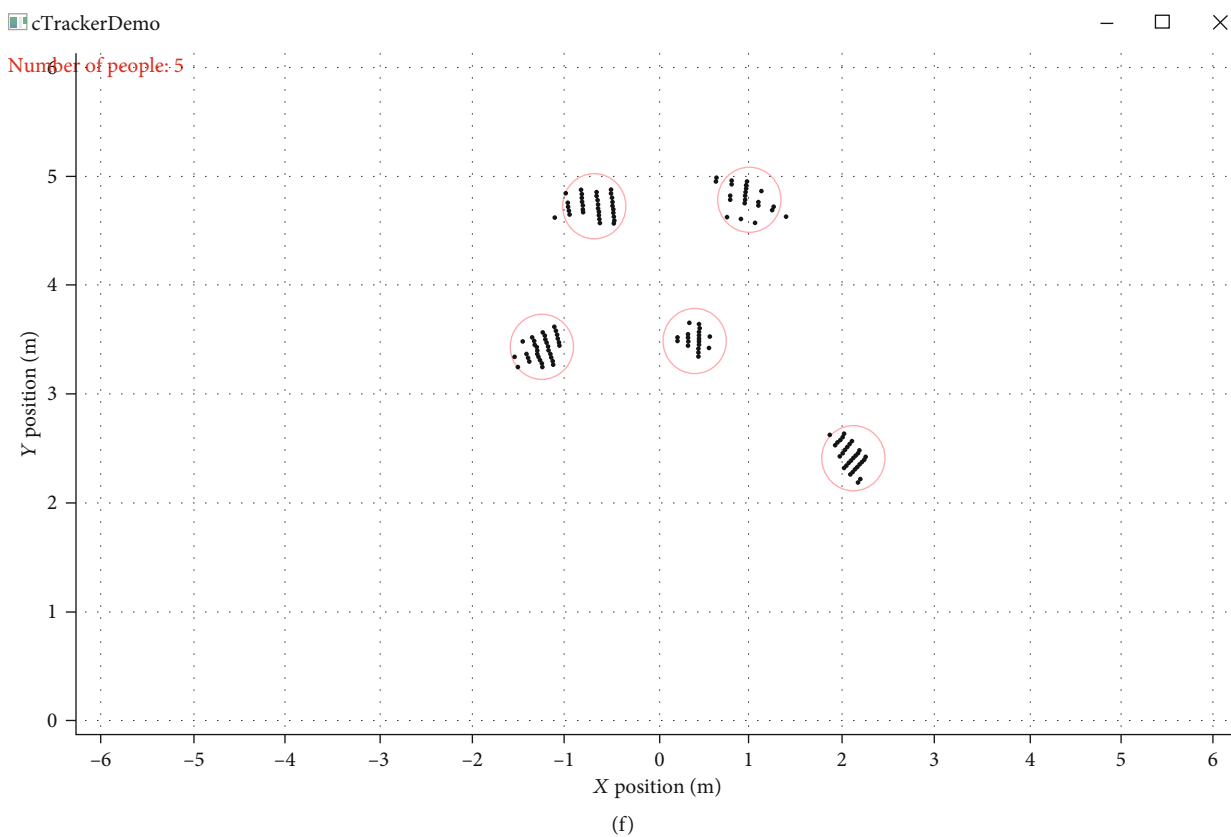
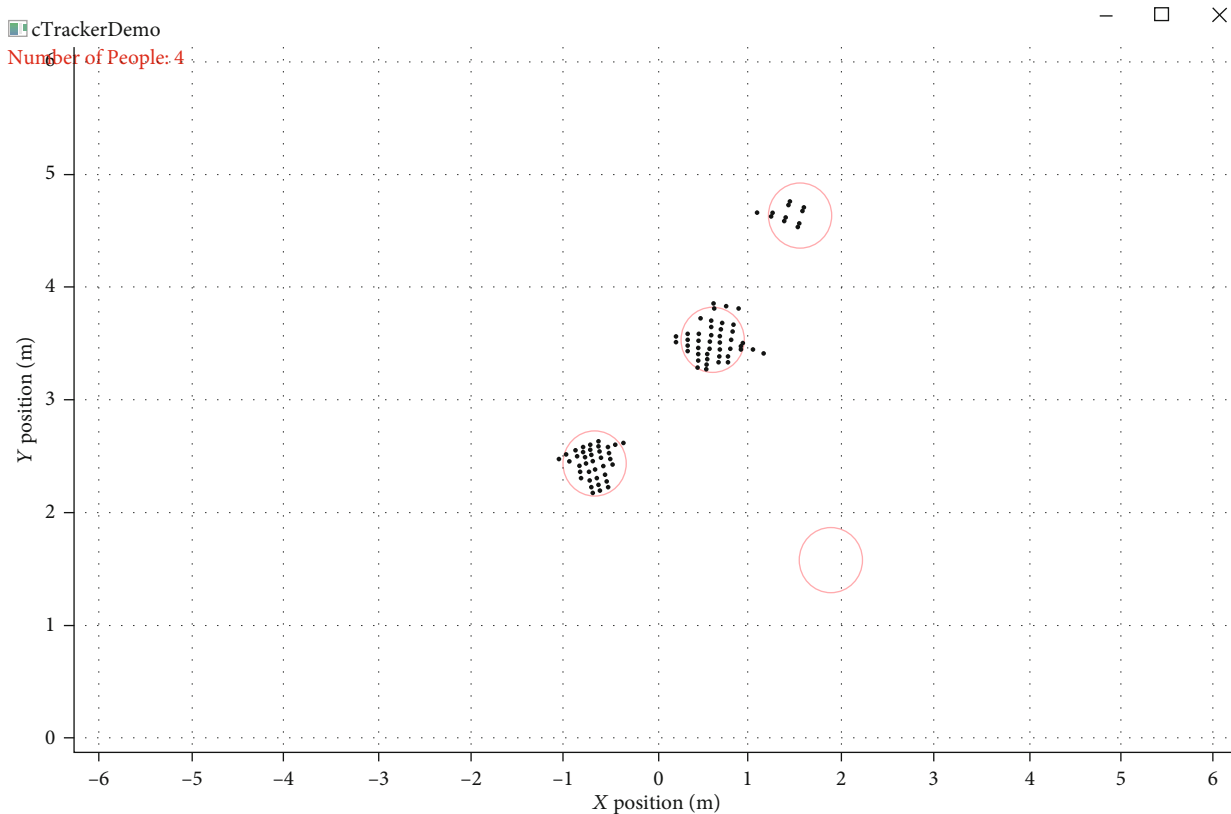


FIGURE 10: cTracker GUI and camera screenshots at the experimental sites: (a) 2 people sitting; (b) 4 people walking; (c) large group; (d) 2 people sitting; (e) 4 people walking; (f) large group.

ground truth information and recording. Figure 7 shows the sensor setup at the different data collection sites. We use data from the three sites to evaluate the methods and algorithms described in the previous sections.

3.2. Evaluation of the Clustering and Referencing. To evaluate the two clustering and referencing algorithms, we tested and compared DBmeans against DBmedoids using a wealth of data obtained from the one experiment site. Experiments were conducted simulating various indoor activities. Data were recorded simultaneously using the TI sensor as well as a video camera, which was used to gather ground truth data. The room was selected to maximize the full range of the sensor. A 6 m by 6 m grid was drawn on the floor to contain the experiment within the sensor range, which allowed us to control when people entered and left the site. The walking activities were selected to test the clustering capabilities of the sensor and to model real indoor scenarios.

Figure 8 shows an example frame of using the DBmeans and DBmedoids algorithms separately. As can be seen, for DBmeans, the centroids are reference locations of each cluster, and for the DBmedoids, the centroids are the real reference points of each cluster. Table 1 also shows the comparison between the DBmeans and the DBmedoids in terms of average misclustering rate and processing time (per frame) using the same total number of data sample frames. In comparison, DBmeans achieves a better average accuracy with 84.75% than DBmedoids with 82.70%. Additionally, DBmeans has a much lower processing time than DBmedoids. Hence, we choose DBmeans as the clustering algorithm of our system.

The density-based clustering algorithm we designed for this task can manage variable cluster densities. Moreover, this algorithm can also handle noise as well as DBSCAN.

3.3. Evaluation of the Tracking. To evaluate the RKF, we compared the tracking accuracy and the processing time of our method to EKF, which TI used.

For the RKF weighting matrix initialization and optimization, we ran through various options and got the best performing combinations. The weighting matrices of the RKF can be initialized as follows:

$$P_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, Q = \begin{pmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{pmatrix}, R = I. \quad (5)$$

By contrast, the EKF is employed to track the same objects. Figure 9 shows the filter results between RKF and EKF using the experiment data set, and Table 2 shows the comparison of the Root Mean Squared Error (RMSE) and process timing (total frames) between the RKF and EKF.

As can be seen, both EKF and RKF can estimate unmeasurable system states and smooth out the process/measurement noise very well. However, in terms of algorithmic

TABLE 3: Comparison between the number of people and time per frame.

Data set	Total frames	Number of people	Time per frame (ms)
RKF	245	3	62.5
EKF	245	3	281.3

complexity and time consumption, the RKF is much more lightweight than the EKF of TI since the RKF does not need to perform coordinate system conversion and calculate the Jacobian matrix which contribute a lot of additional computational load to the system.

3.4. Evaluation of the Merged Process. To evaluate the merged process for scalability and portability, we merged all the algorithms into a tracking system called centroids-Tracker (cTracker) to parse and present the raw point cloud data in real time. Proof of concept for real-time application on a portable embedded platform was demonstrated using a Raspberry Pi 4. This feature's challenge is to design the algorithm to be lightweight enough, such that the processing time is less than each frame's duration. It was done by minimizing the algorithms' timing complexity and writing the program in python with efficient libraries, however, with the limitation of fewer libraries being available on the Raspberry Pi. Efficient code strategies include using lightweight libraries such as NumPy. It provided us with a very quick run time.

Figure 10 presents part of the objects and the camera ground truth. The black points represent the raw point data returned by the mmWave sensor device at experimental sites, and the colored circles represent the clustered and tracked people. As can be seen, all movements, including the walking/standing movements of people, were tracked and represented.

Apart from the code successfully executing on the Raspberry Pi 4, the embedded application's performance in real time also depends on the time complexity of algorithms. Table 3 shows the average processing time (per frame) from a different number of people with data samples between 1 and 5 people. As expected, an increase in the number of people increases the number of points to be processed. More importantly, run time with five people (a high processing load) is below the 50 ms (the frame rate of the mmWave sensor) constraint ensuring consecutive frames are not missed. Besides, our cTracker can track each person correctly, even with some radar measurement data lost (see Figures 10(b) and 10(e)).

The obvious benefit of the algorithm is the Kalman filter's implementation, as no measurement data input would result in the Kalman filter predicting the missing people until they reappear. Besides, if people's data disappears for long periods, the Kalman filter slowly moves people under their predicted velocity, as predicted using the constant velocity model. The prediction eventually estimates the person as having left the room.

3.5. Comparison with TI System. Compared with the misclustering TI system, Figure 11 shows the average misclustering

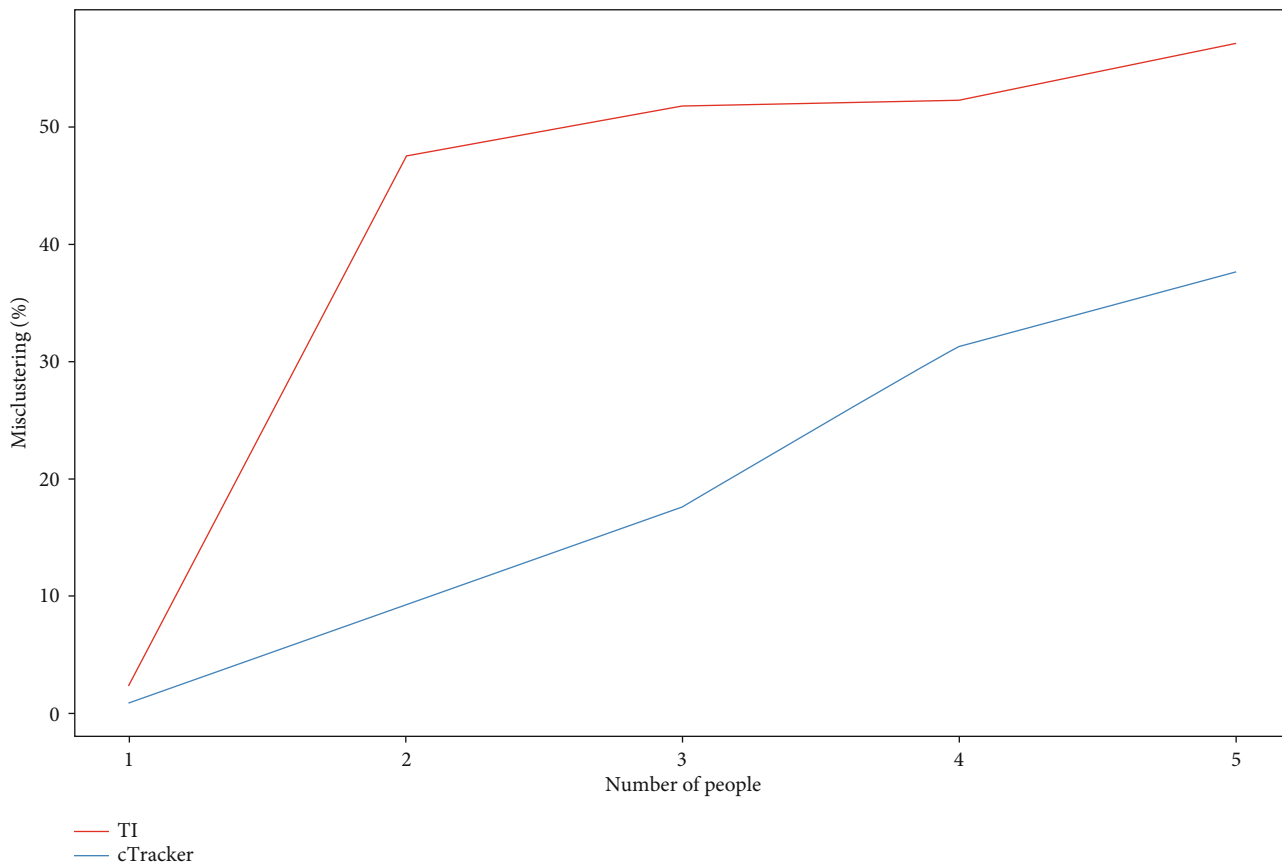


FIGURE 11: Misclustering_TI_vs_cTracker.

TABLE 4: Marked ground truth for tracking accuracy comparison.

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

rate for different numbers of people (total 12917 frames). As can be seen, our system's misclustering rate is much lower than TI's between 1 and 5 people data sets. However, it also displays that both general trends are increasing as the number of people increases. Since the number of people increases, a higher proportion of objects begin occluding each other, leading to a rise in errors. Additionally, missing data from the sensor is another significant reason for increasing the misclustering rate between both systems.

For tracking accuracy comparison, three data sets were collected from a person walking at the position-known location. We then ran those data sets through both our system and the TI system and calculated the RMSE in the X and Y directions. The location coordinate from the sensor is shown in Table 4. Table 5 shows that our system's average position error was 0.2992 meters in location 1, 0.3271 meters in location 2, and 0.3171 meters in location 3. In comparison, the TI system's average position error was 0.3283 meters, 0.3116 meters, and 0.3343 meters, respectively. The TI system was relatively more accurate only at location 2.

TABLE 5: RMSE comparison between two systems.

Data set	Total frames	Systems	RMSE X	RMSE Y	RMSE
Location 1	357	TI	0.3030	0.3518	0.3283
Location 1	357	cTracker	0.2630	0.3316	0.2992
Location 2	163	TI	0.2482	0.3641	0.3116
Location 2	163	cTracker	0.2496	0.3894	0.3271
Location 3	126	TI	0.4331	0.1895	0.3343
Location 3	126	cTracker	0.4218	0.1523	0.3171

4. Conclusion

In this paper, the indoor people detection and tracking system is designed based on the proposed data process algorithms. Our methodology processed in the order of static clutter removal, clustering into clusters, and referencing to identify the centroids, then tracking the centroids by using a recursive Kalman filter (RKF). The experiments are set up at three different data collection sites modelling various indoor scenarios. Comparing with the TI system, our system can detect and track each object more accurately. The processing pipeline cycle is under 50 ms (per frame), which can work in real time on an embedded platform such as a Raspberry Pi. Our future work consists of data fusion from multiple mmWave radar sensors to increase the useful field

of view of the system and accuracy. Moreover, we will also use deep learning approaches for tracking and classifying various species objects.

Data Availability

The data used to support the findings of this study can be freely accessed at <https://github.com/has-c/Occupancy-Detection/tree/master/Data>. The mmWave radar sensor product is obtained from <https://www.ti.com/tool/IWR1642BOOST#technicaldocuments>.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

The authors wish to acknowledge the technical support of the University of Shandong Ying Cai (Jinan, China). This research was funded by the Key Research and Development Project of Shandong Province, China, Grant number 2015GGX101048.

References

- [1] E. Yavari, C. Song, V. Lubecke, and O. Boric-Lubecke, "Is there anybody in there?: Intelligent radar occupancy sensors," *IEEE Microwave Magazine*, vol. 15, no. 2, pp. 57–64, 2014.
- [2] A. Santra, R. V. Ulaganathan, and T. Finke, "Short-range millimetric-wave radar system for occupancy sensing application," *IEEE Sensors Letters*, vol. 2, no. 3, pp. 1–4, 2018.
- [3] G. Monaci and A. V. Pandharipande, "Passive infrared sensor system for position detection," US Patent App. 10/209, 2019, pp. 124.
- [4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [6] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu, "Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 68–87, 2019.
- [7] X. Guo, B. Liu, C. Shi, H. Liu, Y. Chen, and M. C. Chuah, "WiFi-enabled smart human dynamics monitoring," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, p. 16, Delft, Netherlands, November 2017.
- [8] Z. Peng and C. Li, "Portable microwave radar systems for short-range localization and life tracking: a review," *Sensors*, vol. 19, no. 5, p. 1136, 2019.
- [9] C. Iovescu and S. Rao, "The fundamentals of millimeter wave sensors," *Texas Instruments, SPYY005*, vol. 1, pp. 1–9, 2017.
- [10] S. H. Ryu and H. J. Moon, "Development of an occupancy prediction model using indoor environmental data based on machine learning techniques," *Building and Environment*, vol. 107, pp. 1–9, 2016.
- [11] 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 and Buildings*, vol. 158, pp. 1139–1151, 2018.
- [12] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020.
- [13] Y. Benezeth, H. Laurent, B. Emile, and C. Rosenberger, "Towards a sensor for detecting human presence and characterizing activity," *Energy and Buildings*, vol. 43, no. 2-3, pp. 305–314, 2011.
- [14] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: counting people without people models or tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, Anchorage, AK, USA, June 2008.
- [15] K. Garcia, "Bringing intelligent autonomy to fine motion detection and people counting with TImmWave sensors," *Texas Instruments*, vol. 1, pp. 1–9, 2018.
- [16] X. Wang, L. Kong, F. Kong et al., "Millimeter wave communication: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1616–1653, 2018.
- [17] T. Wei and X. Zhang, "mTrack: high-precision passive tracking using millimetre wave radios," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 117–129, Paris, France, September 2015.
- [18] J. Palacios, G. Bielsa, P. Casari, and J. Widmer, "Single- and multiple-access point indoor localization for millimeter-wave networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1927–1942, 2019.
- [19] M. Livshitz, "Tracking radar targets with multiple reflection points," TI internal document, 2017.
- [20] M. Ester, H. P. Kriegel, J. Sander, X. Xu et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," *Knowledge Discovery in Databases*, vol. 96, pp. 226–231, 1996.