Indoor Detection and Tracking of People Using mmWave Sensor

Xu Huang
Department of Electrical and Computer Engineering
The University of Auckland
Auckland 1023, NewZealand
xhua559@aucklanduni.ac.nz
Faulty of Information Engineering
The University of Shandong Ying Cai
Shandong, Jinan, 250104, China
huangxu@ycxy.com

Hasnain Cheena, Abin Thomas, Sing Kiong Nguang*
Department of Electrical and Computer Engineering
The University of Auckland
Auckland 1023, NewZealand
sk.nguang@auckland.ac.nz

***Abstract*—This paper proposes a novel indoor people detection and tracking system based on the Millimeter-wave (mmWave) radar technology. The system framework is firstly presented. Two efficient clustering algorithms are used to cluster and identify people. Then the recursive Kalman Filter (RKF) tracking algorithm is applied to track multiple people simultaneously. The system is implemented based on the proposed approach and compared with the existing solution provided by Texas Instruments (TI). The results show that the proposed system outperformed the existing system in terms of tracking accuracy, computation time, and scalability.**

***Index Terms*—Millimeter Wave, Radar, Detection, Clustering, Tracking**

I. INTRODUCTION

Indoor detection and tracking of people is a useful solution to the problems such as energy assignment, health and safety, etc. [1]. Studies show that indoor detection and tracking system can reduce energy usage for lighting and HVAC systems 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 are falling. 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 measurement, such as passive infrared (PIR), optical cameras, LIDAR, WiFi [3], and 10GHz-to-24GHz microwave. However, all these technologies have challenges in terms of accuracy, privacy, environmental robustness, and system complexity [4]. Motivated by this, this research chose the onboard Millimeter-wave (mmWave) radar sensor (IWR1642BOOST) [5] as the sensing technology [6]. mmWave is a remote wireless sensing technology that has raised lots of attention from both academia and industry due to its exceptional advantages. Compare 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 as well as application guidance using mmWave sensor for the indoor applications.

Ongoing research in object detection and tracking data process technologies are mainly focused on vision-based methods, such as convolutional neural network [7]. There are currently only limited studies using mmWave radar data for indoor detection and tracking of people. In [8], Wei set up a new high-precision passive tracking method (mTrack), and used highly-directional 60GHz millimeter-wave radios 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. The most related work is people counting and tracking using mmWave radar sensor by Texas Instruments (TI) [9]. However, its accuracy is questionable since only DBSCAN [10] is used to clustering the varying density data. Moreover, its portability and scalability are limited due to the use of Extend Kalman Filter (EKF) to convert the polar measurement to Cartesian coordinates. The conversion is taken for the ease of use, yet it brings additional computation load and the process noise.

This paper includes two main contributions. Firstly, we present a systematic approach to the detection and tracking of people indoors by using a mmWave radar sensor. The two efficient clustering algorithms can provide high accuracy and shallow processing time; the RKF tracking algorithm performs much better than the EKF in terms of algorithmic complexity and time consumption. Moreover, a fast indoor people detection and tracking system was designed based

This work was supported in part by the National Natural Science Foundation, China, under Grant 61501284, 51877060, Key Research and Development Project in Shandong Province, China, under Grant 2015GGX101048, and National Statistical Science Research Project, China, under Grant 2019LY82.

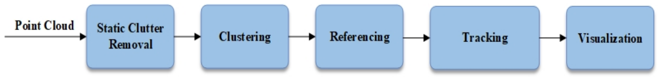


Fig. 1: Framework of the Data Process.

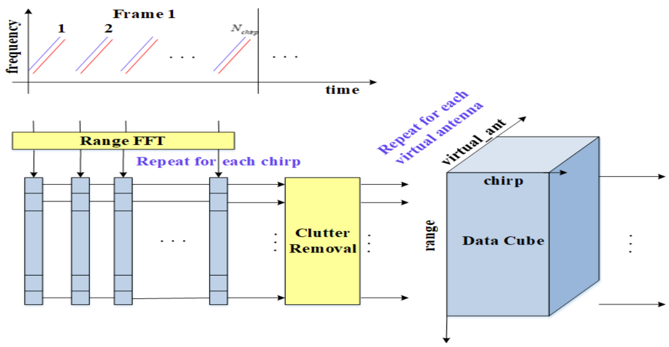


Fig. 2: Structure of Static Clutter Removal Algorithm.

on our proposed algorithms. By compared the results to the commercially available system from TI, the results show the method is faster, more accurate, and less weighted than the TI system.

II. Methodology

A. System Framework

The flow chart in Fig 1 depicts the systematic approach used in this paper for processing and analyzing mmWave sensor data. In this paper, we mainly focus on the clustering and referencing algorithms, tracking algorithm, and timing analysis of the merged approach, then compare it to the method of TI.

B. Static Clutter Removal

Static clutter removal model aims to exclude as many as possible the static points from the background. It requires range information since it filters out non-range changing (static) objects from the scene. Fig 2 demonstrates the static clutter removal algorithm with 3 steps.

Step 1: Range processing performs FFT on ADC samples per antenna per chirp. FFT output is a set of range bins;

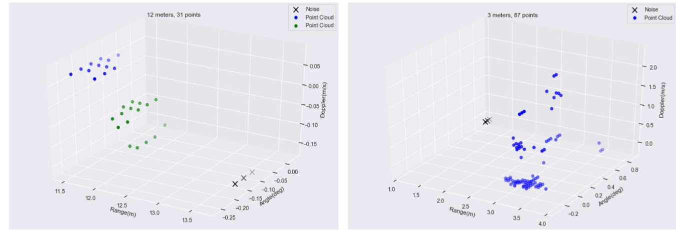
Step 2: Perform static clutter removal by subtracting the estimated DC component from each range bin;

Step 3: Range processing results in local scratch buffers are EDMA to the radar data cube with transpose.

C. Clustering and Referencing

The clustering stage aims to identify the number of people in a scene, and since a single centroid is needed to track each person, a referencing process is required.

1) Clustering: Due to the mmWave sensor field of view, the density of collected data varying from time and distance against the sensor. For example, the closer the people 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 the smaller size objects. To demonstrate, Fig 3 shows the different total number of points (cluster density) of



(a) Data Points at 12 Meters from mmWave Sensor.

(b) Data Points at 3 Meters from mmWave Sensor.

Fig. 3: Comparison of the Numbers of Clustered Point at Different Distances.

people located at different distances against the sensor. There are only 31 points off two people around 12m away from the sensor, while another person who is only 3m away from the sensor, which collected 87 points.

We implemented two density-based clustering algorithms that both could handle varying density points data, and figure out how many people within the space at a given time. The two algorithms description and the comparison between them will be further discussed in more depth below.

2) Referencing: After clustering, all detected people represented by clusters. A reference point on the X, Y plane, needs to be found to locate the position of each cluster. This reference point will later use 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 use as the reference point.

The two density-based clustering 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 miss classification rate. The two algorithms are presented as Algorithm 1 and Algorithm 2.

3) Comparison: To evaluate the two clustering algorithms, we tested and compared DBmeans against DBmedoids using a wealth of data obtained from the experiment sites. Fig 4 shows an example frame of using 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 I also shows the comparison between the DBmeans and the DBmedoids in terms of average miss-clustering 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 perform much lower processing time than DBmedoids. Hence, we choose DBmean as the clustering algorithm of our system.

D. Tracking

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)

Algorithm 1 DBmeans

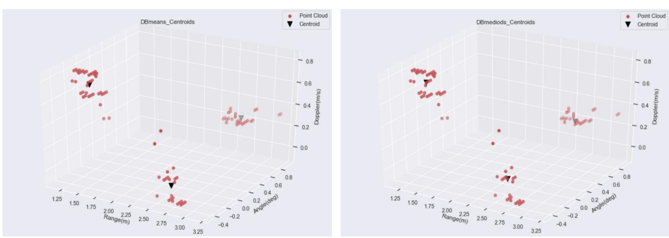
Data:

frame stream dataset - tlv
searching distance - $maxDistance$
minimum number of points to create dense region - $minClusterSize$

Results:

clusterResult, clusterNum, and mean of each cluster (reference location)

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's 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 an outlier is added, change that point's status from outlier to border point.
6: Repeat step 2-5 until all points are either assigned to a cluster or designated as an outlier.
7: Calculate the mean of each cluster.



(a) DBmeans Algorithm for Reference Centroids (b) DBmediods Algorithm for Reference Centroids

Fig. 4: Example frame of the Reference Centriods of DBmeans and DBmedoids Algorithms.

[11] 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. Then we decided to recursively calculate the error covariance matrix and Kalman Gain in each update stage, then we could get a more accurate update, compared to a static Kalman gain.

The 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 also can improve accuracy by avoiding the EKF's process errors caused by linearization by keeping computation in the polar coordinates, which is illustrated in Fig 5. Single reflection point at time n. Multiple reflection points represent real-life radar objects. Each point is represented by range r ($R_{min} < r < R_{max}$), angle θ ($-\theta_{min} < \theta < \theta_{max}$), 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

Algorithm 2 DBmedoids

Data:

frame stream dataset - tlv
searching distance - $maxDistance$
minimum number of points to create dense region - $minClusterSize$

Results:

clusterResult, clusterNum, and medoids of each cluster (reference point or centroid)

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's 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 an outlier is added, change that point's status from outlier to border point.
6: Repeat step 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 medoid.
8: Assign each of the remaining point (non-mediod) in every cluster represented by the nearest medoid.
9: Randomly select a non-medoid 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 non-medoid points caused by the swap; If $S < 0$ then swap O_j with O_{random} to form the new medoid.
11: Repeat step 8-10, until no change.

TABLE I: Comparison Between the DBmeans and the DBmedoids Algorithms

Algorithms	Total Frames	Mis-Clustering	Time (ms)	Accuracy
DBmeans	341	52	29.11	84.75%
DBmedoids	341	59	121.67	82.70%

coordinates for the best view.

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

$$x_k = [r\ \dot{r}\ \theta\ \dot{\theta}]^T \tag{1}$$

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

$$x_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + Q \tag{2}$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_k + R \tag{3}$$

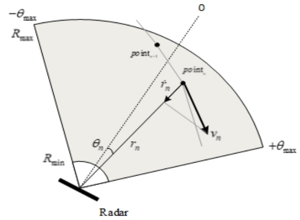


Fig. 5: Radar Geometry in 2D.

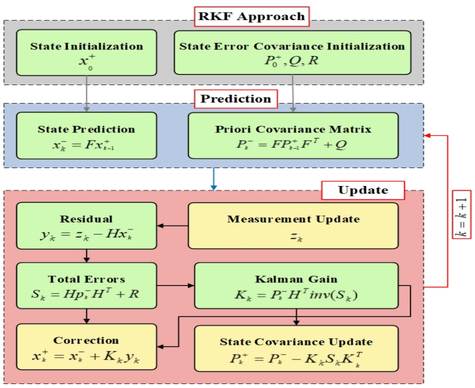


Fig. 6: Implementation Flowchart of the Proposed RKF.

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.

An implementation flowchart of the proposed RKF algorithm summarized in Fig 6. 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^+. By recalculating the Kalman Gain and error covariance, it 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.

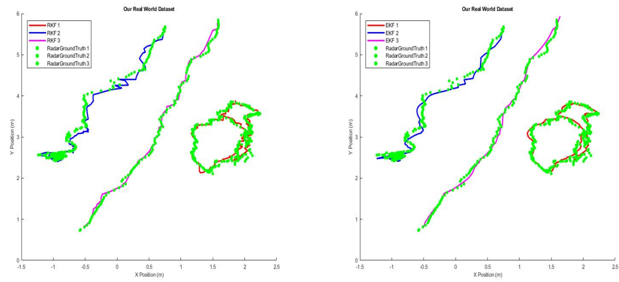
2) Comparison with Extended Kalman Filter: To evaluate the RKF, we compared the tracking accuracy and the processing time of our method to EKF, which TI used.

For the RKF weighting matrices 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:

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}, R = I \tag{4}$$

By contrast, the EKF is employed to track the same objects. Fig 7 shows the filter results between RKF and EKF use experiments dataset and the table II shows the comparison of 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. But in terms of the algorithmic complexity and time consumption, the RKF 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 contributes a lot of additional computational load to the system.



(a) RKF

(b) EKF

Fig. 7: RKF vs EKF

TABLE II: 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

Algorithm 3 Simplified GNN

Data:
centriods - frame stream dataset after clustering

Results:
currentFrame - contains the centroid information for that frame

1: Calculate the distances between each old centroids (objects) and new centriods (observations) in polar coordinate system (e.g. d1, d2, d3 and d4). $d^2 = {r_1}^2 + {r_2}^2 - 2r_1r_2cos(\theta_1 - \theta_2)$
2: Find the one with the globe 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 step 2-3 until all unassociated new centroids and objects are associated (e.g. associate Track 1 and new centroid 2).

3) 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 recursive Kalman filter together, able to work on multiple objects. The globe nearest neighbor (GNN) data association algorithm, which is used in our system, is a simplified version and based on the centroids data after the clustering and referencing step. The simplified GNN diagram and algorithm are shown in Fig 8 and algorithm three.

After GNN is processed, the associated centroids can be passed through the update step of the RKF to be a multi objects tracker.

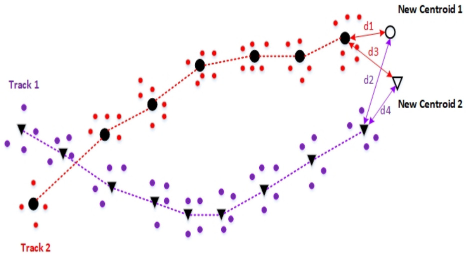


Fig. 8: Simplified Global Nearest Neighbor.

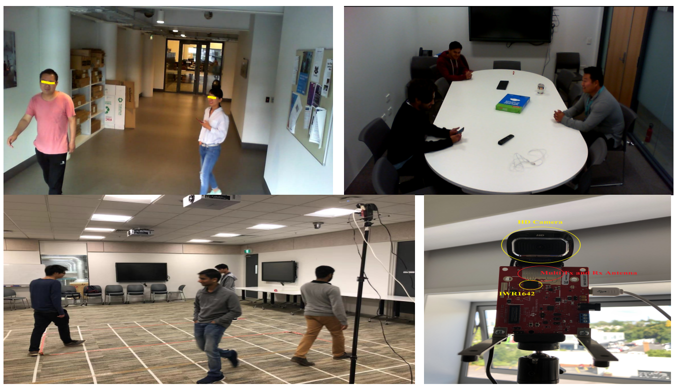


Fig. 9: Various Data Collection Sites (Site 1: Corridor, Site 2: Meeting Room, Site 3: Seminar Room) and Experimental Setup

III. Experiment and Evaluation

A. Experiment Setup

To evaluate the performance of our algorithms, we set up experiments at three different data collection sites around the University of Auckland Newmarket campus to modeling various real indoor scenarios.

For each of the data collection sites, the mmWave sensor mounted to a tripod and elevated to a height of 1.8 to 2 m. The sensor 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°, and 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 ground truth information and recording. Fig 9 shows the sensor set up at the different data collection sites. We use the collecting data from the three sites extensively to evaluate the methods and algorithms described in the previous sections.

B. Evaluation of the Merged Process

To evaluate the merged process, we merged all the algorithms into a tracking system called centroids-Tracker (cTracker) to parsing and presenting the raw point cloud data in real-time. Fig 10 presents part of the objects, and the camera ground truth, in which the black points represent the raw points 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,

(a) 2 People Sitting (b) 4 People Walking (c) Large Group

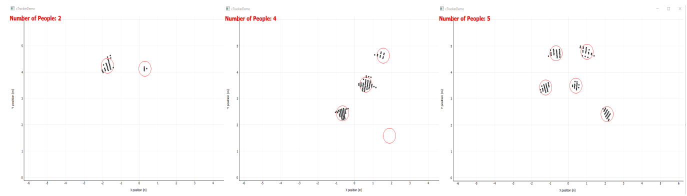


Fig. 10: cTracker GUI and Camera Screenshots at the Experimental Sites

TABLE III: Comparison Between the Number of People and the Time per Frame

DataSet	Total Frames	Number of People	Time per Frame (ms)
2PeopleSitting	723	2	17.39
4PeopleWalking	866	4	33.93
LargeGroup	1169	5	38.46

including the walking/standing movements of people, were tracked and represented. Table III shows the average processing time (per frame) from the different number of people data samples between 1-5 people. As can be seen, all of the dataset processing are below 50ms (the frame rate of the mmWave sensor), ensuring consecutive frames are not missed. Besides, our cTracker can track each people correctly, even with some radar measurement data lost (see Fig 10 (b)).

C. Comparison with TI System

Compared with the TI system for the miss-clustering, Fig 11 shows the average miss-clustering rate for different numbers of people (total 12917 frames). As can be seen, our system's miss-clustering rate is much lower than TI's between 1 to 5 people dataset. 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 the increase of the miss-clustering rate between both systems.

For tracking accuracy comparison, three datasets were collected from a person walking at the position-known location. Then, we ran those datasets through both our system and the TI system and calculating the RMSE in X, Y directions. The location coordinate from the sensor are shown in Table IV. Table V shows the average position error of our system were 0.2992 metres in location 1, 0.3271 metres in location 2, and 0.3171 metres in location 3. In comparison, the average position error of the TI system was 0.3283 meters, 0.3116 meters, and 0.3343 meters, respectively. The TI system was relatively more accurate only at location 2.

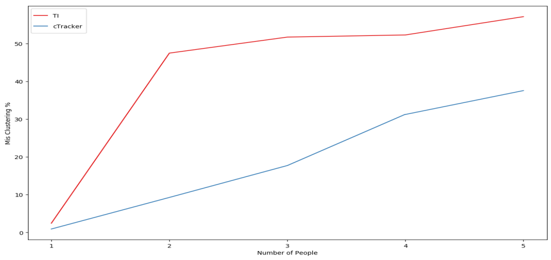


Fig. 11: Miss-Clustering vs Number of People.

TABLE IV: 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

TABLE V: RMSE Comparison Between Two Systems

DataSet	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

IV. Conclusion

In this paper, an 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 modeling various indoor scenarios. By comparing to the TI system, our system can detect and track each object more accurately, and the cycle of the processing pipeline is under 50 ms (per frame), which is a fast and small-sized real-time object tracking system. Our future work consists of data fusion from multiple mmWave radar sensors to increase the useful field of view of the system as well as accuracy. And to use deep learning approaches for tracking and classify various species objects.

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] 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*. ACM, 2017, p. 16.

[4] C. Iovescu and S. Rao, "The fundamentals of millimeter wave sensors," *Texas Instruments, SPYY005*, vol. 1, pp. 1–9, 2017.

[5] K. Garcia, "Bringing intelligent autonomy to fine motion detection with mmwave sensors," *Texas Instruments*, vol. 1, pp. 1–9, 2018.

[6] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon, and G. Chen, "Millimeter wave communication: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1616–1653, 2018.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] T. Wei and X. Zhang, "mtrack: High-precision passive tracking using millimeter wave radios," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 117–129.

[9] T. Instruments, "People counting and tracking using mmwave radar sensor with sub-1 ghz reference design," 2018, pp. 1–35.

[10] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[11] I. Reid, "Estimation ii," in *Hilary Term*, 2001, pp. 1–44.