

Development of Intelligent Multimodal Traffic Monitoring using Radar Sensor at Intersections

Siyang Cao, Ph.D.

Yao-jan Wu, Ph.D.

Feng Jin, Ph.D.



Development of Intelligent Multimodal Traffic Monitoring using Radar Sensor at Intersections

Final Report

NITC-RR-1296

by

Siyang Cao, Ph.D., University of Arizona
Yao-jan Wu, Ph.D., P.E., University of Arizona
Feng Jin, Ph.D., University of Arizona

for

National Institute for Transportation and Communities (NITC)
P.O. Box 751
Portland, OR 97207



November 2021

Technical Report Documentation Page			
1. Report No. NITC-RR-1296	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Development of Intelligent Multimodal Traffic Monitoring using Radar Sensor at Intersections		5. Report Date November 2021	
		6. Performing Organization Code	
7. Author(s) Siyang Cao, orcid.org/0000-0001-9593-265X Yao-Jan Wu, orcid.org/0000-0002-0456-7915 Feng Jin, orcid.org/0000-0002-2985-3328		8. Performing Organization Report No.	
9. Performing Organization Name and Address The University of Arizona Department of Civil & Architectural Engineering & Mechanics 1209 E. Second St. P.O. Box 210072 Tucson, AZ 85721		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. NITC 1296	
12. Sponsoring Agency Name and Address National Institute for Transportation and Communities (NITC) P.O. Box 751 Portland, OR 97207		13. Type of Report and Period Covered Final Report 08/01/2019-08/31/2021	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract Multimodal traffic monitoring is critical for improving mobility and safety at intersections with potential conflicts among various modes of transportation. Traditional traffic monitoring approaches utilizing cameras cannot work reliably during the night and under hazardous weather conditions. We propose to build a new intelligent multimodal traffic monitoring device using the low-cost mmWave radar. The proposed device can reliably distinguish different modes (such as buses, pedestrians, bicyclists, trucks, motorcycles, etc.), and determine the counts, speed, and moving directions of every single target in an urban environment under various lighting and weather conditions. In the study, a low-cost prototype system will also be built and tested for data collection at a major intersection near the University of Arizona. The proposed research can build a foundation for commercializing a new traffic monitoring device through the project.			
17. Key Words Multimodal, traffic monitoring, radar, intersections		18. Distribution Statement No restrictions. Copies available from NITC: www.nitc-utc.net	
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 22	22. Price

ACKNOWLEDGEMENTS

This study was funded by the National Institute for Transportation and Communities (NITC; grant number 1296) a U.S. DOT University Transportation Center. The authors would like to also thank the Tucson Department of Transportation for funding support.

DISCLAIMER

The contents of this report reflect the views of the authors, who are solely responsible for the facts and the accuracy of the material and information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation University Transportation Centers Program and Tucson Department of Transportation in the interest of information exchange. The U.S. assumes no liability for the contents or use thereof. The contents do not necessarily reflect the official views of the U.S. Government. This report does not constitute a standard, specification, or regulation.

RECOMMENDED CITATION

Cao, S., Wu, Y. Jin, F., *Development of Intelligent Multimodal Traffic Monitoring using Radar Sensor at Intersections*. NITC-RR-1296. Portland, OR: Transportation Research and Education Center (TREC), 2021.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	4
1.0 INTRODUCTION.....	5
1.1 PROJECT OVERVIEW	5
1.2 TECHNICAL BACKGROUND	5
1.3 TECHNICAL REVIEW.....	7
1.3.1 Image Segmentation.....	7
1.3.2 LiDAR Point Cloud Segmentation	8
1.3.3 Radar Point Cloud Segmentation	8
1.3.4 Basics of Multivariate Gaussian Mixture Model [8].....	9
1.3.5 MmWave Radar Point Cloud in Multimodal Traffic Monitoring.....	11
2.0 PROJECT DETAILS	13
2.1 EXPERIMENT SETUP	13
2.2 REAL-TIME DATA COLLECTION.....	14
2.3 EXPERIMENT RESULTS.....	16
3.0 PROJECT CONCLUSION.....	20
4.0 REFERENCES	21

LIST OF TABLES

Table 1.1: Radar point cloud data	11
Table 2.1: Performance matrices	17

LIST OF FIGURES

Figure 1.1: Segmentation examples in which the color represents the class of object. (a) Image segmentation [4]. (b) 3D (or LiDAR) point cloud segmentation [5]. (c) Radar point cloud segmentation [6]	7
Figure 1.2: The EM algorithm.....	11
Figure 1.3: MmWave radar point cloud example. Red axis: x; Green axis: y; Blue axis: z.	12
Figure 2.1: Experimental setup.	13
Figure 2.2: Real-time data collection for a running sedan. Left is video view; Right is radar view.	15
Figure 2.3: One frame of results. Red point: clutter; Green point: car; Blue point: pedestrian. (a) Prediction. (b) Ground truth.....	16
Figure 2.4: Precision-recall curve.....	18
Figure 2.5: Confusion matrix	18

EXECUTIVE SUMMARY

Intelligent transportation systems (ITS) significantly change our communities by improving the safety and convenience of people's daily mobility. The system relies on multimodal traffic monitoring that needs to provide reliable, efficient and detailed traffic information for traffic safety and planning. Signalized traffic intersections are critical spots for collecting such mixed traffic data because most conflicts and crash occurrences involve multiple transportation modes, such as pedestrians, bicyclists, motorcyclists, and cars. How to reliably and intelligently monitor intersection traffic with multimodal information is one of the most critical topics in intelligent transportation research.

This project investigated a low-cost, low-weight, compact, and reliable monitoring platform. This platform, which incorporates high-resolution millimeter-wave(mmWave) radar and the machine-learning technique to collect multimodal traffic data at intersections, is robust to light and adverse weather conditions. The products of this project consist of 1) a prototype of the proposed multimodal traffic monitoring platform using mmWave radar; 2) the real-world experimental dataset collected by the platform for multimodal traffic; and 3) a demo platform at a road intersection to illustrate the performance in terms of measuring multimodal traffic counts, speeds, and directions.

Our primary goal is to improve multimodal traffic monitoring at intersections. The developed platform can play an important role in providing a reliable and accurate city-wide traffic network. In addition, the outcome of this research can provide useful insight into advanced innovations technologies for developing equitable, healthy, and sustainable communities and smart cities.

1.0 INTRODUCTION

1.1 PROJECT OVERVIEW

Intelligent transportation systems (ITS) significantly change our communities by improving the safety and convenience of people's daily mobility. The system relies on multimodal traffic monitoring that needs to provide reliable, efficient and detailed traffic information for traffic safety and planning. Signalized traffic intersections are critical spots for collecting such mixed traffic data because most conflicts and crash occurrences involve multiple transportation modes, such as pedestrians, bicyclists, motorcyclists, and cars. How to reliably and intelligently monitor intersection traffic with multimodal information is one of the most critical topics in intelligent transportation research.

Technically speaking, in multimodal traffic monitoring we gather traffic statistics for distinct transportation modes, such as pedestrians, cars and bicycles, in order to analyze and improve people's daily mobility in terms of safety and convenience. On account of its robustness to bad light and adverse weather conditions, and inherent speed measurement ability, the radar sensor is a suitable option for this application. However, the sparse radar data from conventional commercial radars make it extremely challenging for transportation mode classification, this is a common difficulty in radar point cloud segmentation research [6]. Thus, we propose to use a high-resolution millimeter-wave (mmWave) radar sensor to obtain a relatively richer radar point cloud representation for a traffic monitoring scenario. Based on a new feature vector, we use the multivariate Gaussian mixture model (GMM) to do the radar point cloud segmentation (i.e., "pointwise" classification), in an unsupervised learning environment. In our experiment, we collected radar point clouds for pedestrians and cars, which also contained the inevitable clutter from the surroundings. The experimental results using GMM on the new feature vector demonstrated a good segmentation performance in terms of the intersection-over-union (IoU) metrics. The detailed methodology and validation metrics are presented and discussed.

1.2 TECHNICAL BACKGROUND

Using traditional radar signal processing, we obtain the position and Doppler information of reflection points from the scene after a suitable detection stage, such as Constant False Alarm Rate (CFAR) processing. The resulting positional representation in 3D space is referred to as a radar point cloud, derived from a similar terminology used for LiDAR point cloud. Radar point cloud segmentation is a point-wise classification, which means it would classify each reflection point into a specific class. Segmentation for data obtained using camera (image or pixel array) and LiDAR (point cloud) have been

continuously and extensively studied, primarily for autonomous driving and machine perception.

Although relatively new, radar point cloud segmentation has also started to garner attention, given its several advantages over the other sensor modalities. Traditional commercial radars offer limited resolutions, in both range and angle, which leads to a very sparse representation of the object from the radar's perspective.

This also implies that segmentation on the sparse data is extremely difficult to model, often yielding sub-par results. On the other hand, camera and LiDAR provide a very dense pixel array and point cloud representation of the scene, respectively, that in turn yields a superior segmentation performance.

The recently emerging millimeter-wave (mmWave) frequency modulated continuous wave (FMCW) radar devices offer range resolution of up to 5 cm on account of an ultra-bandwidth of up to 4 GHz, using carrier frequencies of 60GHz, 77GHz and 80GHz, depending on the area of application. Furthermore, with advanced semiconductor fabrication process, more radio frequency channels are interpreted into a single monolithic microwave integrated circuit (MMIC) chip. This allows compact mmWave radars to provide relatively good angle resolution compared to outdated bulky commercial radars. Several examples of these mmWave FMCW MMIC radar chips include the Texas Instruments AWR1843 [1], NXP TEF810X [2] and Infineon RXS816xPL [3].

With the availability of such high-resolution radars, we can now obtain a relatively richer reflection point cloud representation of a single object, especially in the near range operation (less than 30 meters). Therefore, radar point cloud segmentation could be targeted by utilizing techniques from the traditional image and LiDAR processing domains.

Furthermore, subsequent radar data post-processing, such as object clustering, tracking and classification, could be rebuilt using machine learning and deep-learning architectures, similar to the ones used for images and/or LiDAR data, that have shown to yield very promising results.

Particularly, in multimodal traffic monitoring, sensors need to be employed to (i) estimate the traffic volume of different transportation modes, such as pedestrian, motorcycle and car, and (ii) estimate their average speeds. In order to achieve that, the sensor needs to be robust to operating all day and in any weather condition with the additional capability to accurately estimate the speed of the objects, which makes radars a suitable choice. With the high-resolution relatively dense point cloud representation of each object, classification to a suitable transportation mode can be feasible by using a segmentation approach.

In this report, we use a single high-resolution mmWave radar device to monitor an experimental scene with pedestrians and cars in it, and gather the radar point clouds. We propose to compute a new feature vector for each radar point. Then, we use a

multivariate Gaussian mixture model (GMM) as the decision algorithm to perform the radar point cloud segmentation (i.e., point-wise classification).

1.3 TECHNICAL REVIEW

This section reviews some of the latest segmentation techniques in the application domains of image, LiDAR and radar processing, as shown in Fig. 1.1.

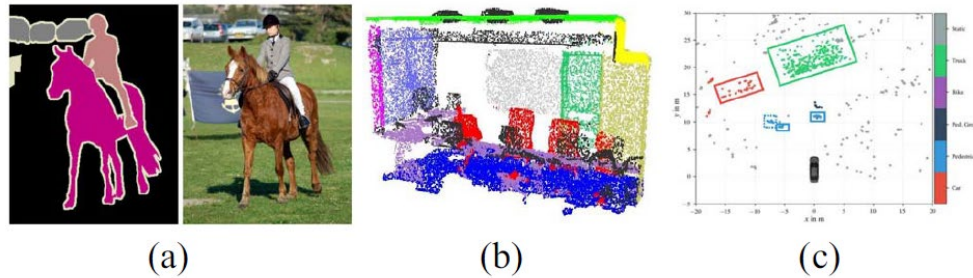


Figure 1.1: Segmentation examples in which the color represents the class of object. (a) In the image segmentation [4], pixels are classified as different objects. (b) In the 3D (or LiDAR) point cloud segmentation [5], points are classified. (c) Similarly, in the radar point cloud segmentation [6], points are also classified.

1.3.1 Image Segmentation

Although image segmentation has had very broad approaches with a long research history, those methods, such as thresholding-based, edge-based and region-based [7], heavily depend on the intensity (in grey or color) scale of each pixel. However, the radar cross section, which is analogous to the intensity in the radar point cloud domain, may be too vague to be used.

On the other hand, the clustering-based methods, such as k-means, Gaussian mixture model (GMM) and density-based spatial clustering of applications with noise (DBSCAN), that realize models to estimate the density or intensity scale of the pixels and could be considered as valid choices for radar point cloud segmentation. Specifically, k-means assigns all the pixels into k clusters by minimizing the sum of the squared distance of all the pixels to its own cluster, as intuitively a cluster is thought of a group of data points whose inter-point distances are small compared to the distances to points outside of the cluster [8]. The GMM models a group of data as a weighted sum of Gaussian distributions, where each distribution accounts for a unique cluster. A cluster, in this case, is formed if all the points obey the same Gaussian distribution [9]. In DBSCAN, a core point is defined if in its neighborhood of a given radius, there are at least a given minimum number of points. Then, the DBSCAN algorithm forms a cluster for all density-reachable points (i.e., each point is within the neighborhood of the core point and all density-connected points; there is a third point from which both of these two points are reachable [9]).

Recently, deep-learning-based approaches have shown very promising results in image segmentation. In [4], the authors proposed a fully convolutional network with end-to-end training on pixel-level labeled images. In [10], the authors proposed the R-CNN: regions with CNN features to first extract the region of interest (ROI) along with CNN features computation, and then to classify the region using a linear support vector machine. The success of supervised deep-learning approaches motivates the researchers to apply it on the LiDAR point cloud segmentation.

1.3.2 LiDAR Point Cloud Segmentation

Each LiDAR point contains the information of 3D position and intensity. With a dense 3D point cloud representation of the object, the authors from Stanford proposed the PointNets family, including the PointNet [5], PointNet++ [11] and Frustum PointNet [12], to learn the 3D spatial feature of the object, which is a pioneering work on directly processing LiDAR point cloud, compared to the other traditional ways that may do the voxelization first and make the data unnecessarily voluminous. The authors first proposed a vanilla PointNet to transform the three-dimensional LiDAR point to the 1024-dimensional space in which the pattern of the different object can be more likely separable, according to the Cover's theorem on the separability of patterns [13]. The basic architecture of the vanilla PointNet consists of multilayer perceptions to learn the feature space transformation in a supervised fashion with numerous labeled point data. And then, the authors devised the T-Net, a simplified vanilla PointNet, to learn the transformation of the object, such as translation, rotation and scaling, so that the entire PointNet architecture can be transformation invariant.

In PointNet++, an extension of the PointNet, the authors introduced (i) the convolution operation with the PointNet as the kernel to learn the local spatial features; (ii) the multiscale and multiresolution grouping to deal with the variation in different areas; and (iii) and the farthest point sampling to sample the points in a more efficient way.

Finally, in Frustum PointNet, the authors first used the typical convolution neural network (CNN) to detect the region of interest (ROI) in the 2D images, and then extracted the frustum of ROI in the 3D point cloud to represent the object followed by a PointNet++ model to do the classification. The PointNet family can do one object classification and scene segmentation.

1.3.3 Radar Point Cloud Segmentation

Although segmentation in the synthetic aperture radar image processing [14] has been studied several years ago, segmentation on the radar point cloud has a very short history. This is on account of the previous real aperture radar's limited resolution, resulting in poor segmentation results, while the synthetic aperture radar has a relatively better resolution.

With the great success of the PointNet family on LiDAR and 3D point cloud processing, researchers have attempted to try it out on radar point cloud, expecting promising results. In [6], the authors first accumulated multiple radar frames to obtain a richer point

cloud, and then applied the Frustum PointNet with some minor adaptations on the 2D radar point cloud and claimed better segmentation results over their previous work [15], in which they used a combination of DBSCAN and long short-term memory (LSTM) network to predict the class for each radar point. And in [16], the authors applied PointNets on the 2D radar point cloud to differentiate the vehicle from clutter with the vehicle bounding box estimation.

However, from our understanding, because the PointNet family is designed for learning the spatial 3D features of the object, it may not have meaningful and practical results on the radar point cloud. As the radar point cloud is still very sparse compared to the LiDAR point cloud, it leads to the loss of some spatial features. Accumulation of multiple radar frames can improve the data. For a high-speed vehicle, however, its radar points would have moved a significant distance just after a few frames so that the accumulation does not make sense. Moreover, the availability of labeled radar point cloud is rare and difficult to gather, so the supervised learning approaches may not be a good option.

1.3.4 Basics of Multivariate Gaussian Mixture Model [8]

Given a set of data points in which each point is a vector, the goal is to classify each point into a single class. We assume there are a total of K classes these points may belong to. For a data point x , given that it belongs to the k -th class (i.e., $c_k = 1$, $k \in \{1, \dots, K\}$), it is assumed to follow a certain multivariate Gaussian distribution as

$$p(x|c_k = 1) = \mathcal{N}(\mu_k, \Sigma_k) = \frac{1}{(2\pi)^2 |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}, \quad (1)$$

where μ_k is the mean and Σ_k is the covariance matrix for the k -th class.

Then a data point with an unknown class should follow a GMM, which is a linear superposition of Gaussian distributions of all the K classes, by the law of total probability, as

$$p(x) = \sum_{k=1}^K p(c_k = 1) p(x|c_k = 1) \quad (2)$$

where the $p(c_k = 1)$, also denoted as π_k , is the prior probability of $c_k = 1$ or x belongs to the class c_k .

If the parameters (i.e. (π, μ, Σ) for all the K classes), are given, so the posterior probability of $c_k = 1$ for a given radar point, by the Bayes' theorem, is

$$\begin{aligned}\gamma(c_k) &= p(c_k = 1|x) \\ &= \frac{p(x|c_k = 1)p(c_k = 1)}{p(x)} = \frac{\pi_k \mathcal{N}(\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mu_j, \Sigma_j)},\end{aligned}\quad (3)$$

where $\gamma(c_k)$ can also be viewed as the responsibility that the class K takes for “explaining” the data point x .

Then, we can use the maximum a posteriori criterion to determine the class of each radar point, that is

$$k = \max_j \gamma(c_j), \quad j \in \{1, \dots, K\}, \quad (4)$$

Thus, the remaining question is how to determine all the parameters in GMM. The expectation maximization (EM) algorithm can be applied as following. Assuming a set of data points $X\{x_1, \dots, x_N\}$ with unknown classes is collected, the optimal parameters (π^o, μ^o, Σ^o) are those to maximize the likelihood function based on the maximum likelihood esn, given by

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^N \ln p(x_n) = \sum_{n=1}^N \ln \sum_{k=1}^K \pi_k p(x_n|c_k = 1), \quad (5)$$

The optimal parameters occur when the partial derivative of the likelihood function with respect to each parameter is zero. Then we have

$$\mu_k^o = \frac{1}{N_k} \sum_{n=1}^N \gamma(c_k^n) x_n, \quad (6)$$

$$\Sigma_k^o = \frac{1}{N_k} \sum_{n=1}^N \gamma(c_k^n) (x_n - \mu_k^o)(x_n - \mu_k^o)^T, \quad (7)$$

$$\pi_k^o = \frac{N_k}{N}, \quad (8)$$

where $N_k = \sum_{n=1}^N \gamma(c_k^n)$ and $c_k^n = 1$ means the n -th point belongs to the k -th class.

So, the EM algorithm, as in Fig. 1.2, will iteratively update the parameters until the convergence of either the parameters or the log likelihood has been achieved.

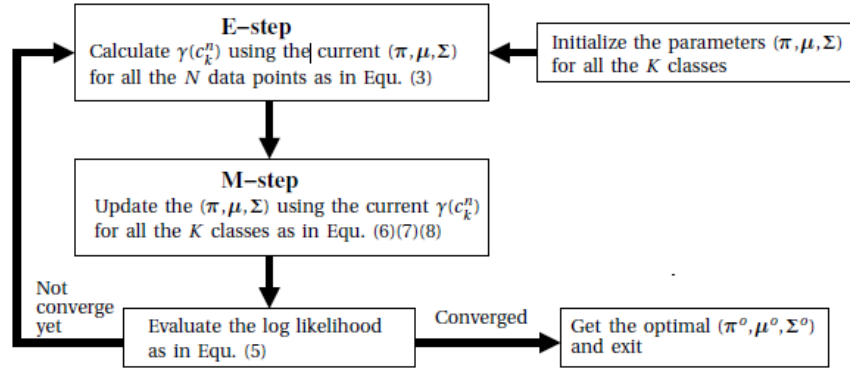


Figure 1.2: The EM algorithm flow chart

1.3.5 MmWave Radar Point Cloud in Multimodal Traffic Monitoring

After the traditional FMCW radar signal processing chain (i.e., range-FFT, Doppler-FFT, angle-FFT, moving target indication (MTI), constant false alarm rate (CFAR), clustering and tracking), we obtain the radar point cloud, in which each point is a vector $x(r, \theta_{az}, \theta_{el}, vD, snr, noise, pX, pY, pZ, vX, vY, vZ)$. Its parameters are listed in Table 1, in which the point data represents the radar measurement of each radar reflection point in the polar coordinate, and the centroid data represents the Kalman filtering based tracking results of the centroid of each tracked object in the Cartesian coordinate.

Table 1.1: Radar point cloud data

Point Data			Centroid Data		
Symbol	Value	Unit	Symbol	Value	Unit
r	range	m	pX	x position	m
θ_{az}	azimuth angle	degree	pY	y position	m
θ_{el}	elevation angle	degree	pZ	z position	m
vD	Doppler velocity	m/s	vX	x velocity	m/s
snr	Signal-to-noise ratio	dB	vY	y velocity	m/s
noise	CFAR window noise	dB	vZ	z velocity	m/s

Then we propose the feature vector $(\Delta x, \Delta y, \Delta z, \Delta D, \sigma)$ for each radar point, where

$$\Delta x = r * \cos(\theta_{el}) * \sin(\theta_{az}) - pX, \quad (9)$$

$$\Delta y = r * \cos(\theta_{el}) * \cos(\theta_{az}) - pY, \quad (10)$$

$$\Delta z = r * \sin(\theta_{el}) - pZ, \quad (11)$$

$$\begin{aligned} \Delta D &= vD - (vX, vY, vZ) \cdot (the_point_direction) \\ &= vD - [\cos(\theta_{el}) * \sin(\theta_{az}) * vX + \\ &\quad \cos(\theta_{el}) * \cos(\theta_{az}) * vY + \sin(\theta_{el}) * vZ], \end{aligned} \quad (12)$$

$$\sigma = 10 \log_{10} r^4 + snr + noise. \quad (13)$$

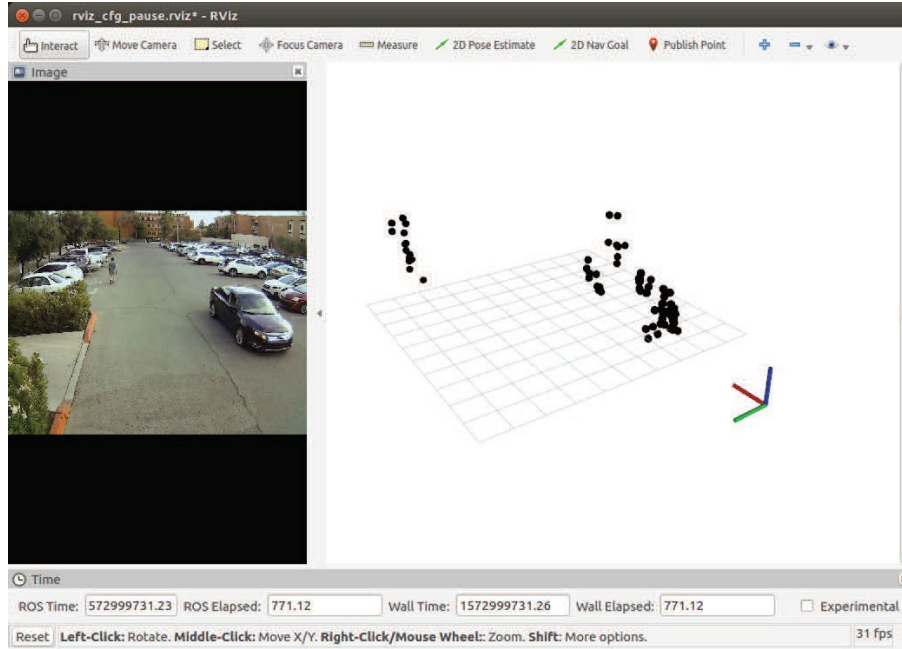


Figure 1.3: MmWave radar point cloud example. Red axis: x; Green axis: y; Blue axis: z. On the left, it's the camera video for ground truth as reference; on the right, it's the radar point cloud in which two clusters are detected for two moving objects, i.e., the walking pedestrian and the running sedan in the image on the left.

As a result, (i) the $(\Delta x, \Delta y, \Delta z)$ is the relative position of each point with respect to the object centroid, and represents the extent of the object body; (ii) the ΔD represents the relative Doppler; (iii) and the σ is the radar cross section in the unit of dBsm. We observe that each point from one kind of object obeys a certain Gaussian distribution with its own mean and variance. In the multimodal traffic monitoring, because the size, speed and reflection coefficient of a pedestrian is less distinguishable than these of a sedan, GMM can be applied for classification between these two. It is the same for the other transportation modes. Fig. 1.3 shows an example of a radar point cloud including a car and a pedestrian from our data collection that will be further described in Section IV. Here we can see the differences between the distributions of points from these two kinds of objects.

2.0 PROJECT DETAILS

2.1 EXPERIMENT SETUP

We used a TI mmWave radar evaluation board AWR1843BOOST [17] to get the radar point cloud, the Nvidia Nano [18] to process the data, and one USB camera for capturing the video as a reference. Fig. 2.1 shows how we collected the experimental data in a parking lot and an intersection. The device was raised up to 3 meters high, and all the data was wirelessly transferred to a laptop for storage.



(a) Parking lot

(b) Intersection

Figure 2.1: Experimental setup that includes one laptop for device control and data storage, one mmWave radar for data collection, one wifi module for data wirelessly transferring and one camera for visual ground truth recording. The mmWave radar, wifi module and camera are bounded together and raised to about 3 meters for the height of a traffic light. (a) Setup in a parking lot for a preliminary experimental testing; (b) Setup in the intersection for a more realistic experimental testing.

With proper FMCW waveform design and the implementation of multiple-input-multiple-output direction-of-arrive algorithm, we achieved about 9 centimeters of range resolution, 0.8 m/s of Doppler resolution, 15 degrees of azimuth angle resolution, and 28 degrees of elevation angle resolution. For now, we only collected the data with two different kinds of transportation modes (i.e., pedestrian and car). The effective radar detection area is up to 15 meters in range and 18 meters in cross-range, for both car and pedestrian. The data may also include the inevitable clutter or noise.

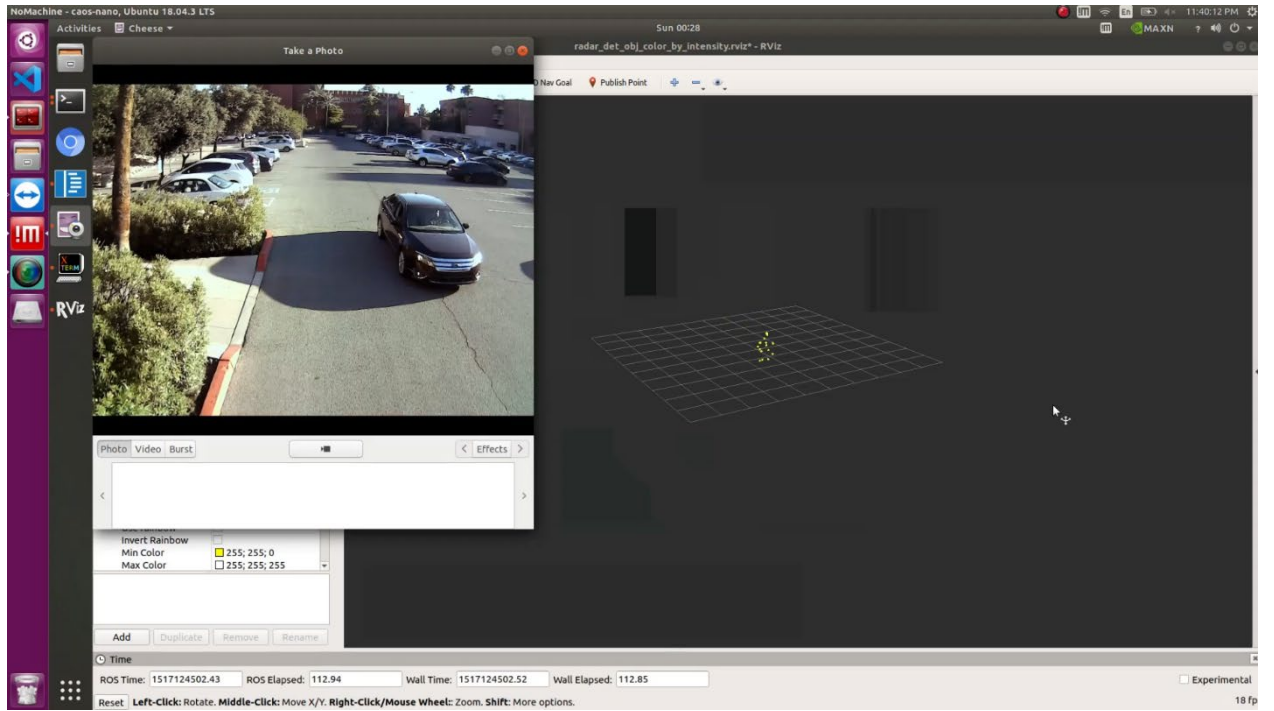
When we set up the device at the intersection, we first set up a wifi router to build a local area network (LAN). After that, for the first time we need to use an external HDMI

LCD screen, keyboard and mouse to configure the Nano to connect to the wifi network. This is one-time work, and next time the Nano will automatically connect to the remembered wifi network. Then we connected the radar to the Nano. Once it's ready, we disconnect LCD, keyboard and mouse from the Nano and raised the Nano and radar to 3 meters high, as shown in the figure above. On the computer that was connected to the same wifi network, we then used the remote software called NoMachine to remotely control the Nano. Until now, we were able to execute Linux commands on Nano remotely to start the radar for continuous data collection.

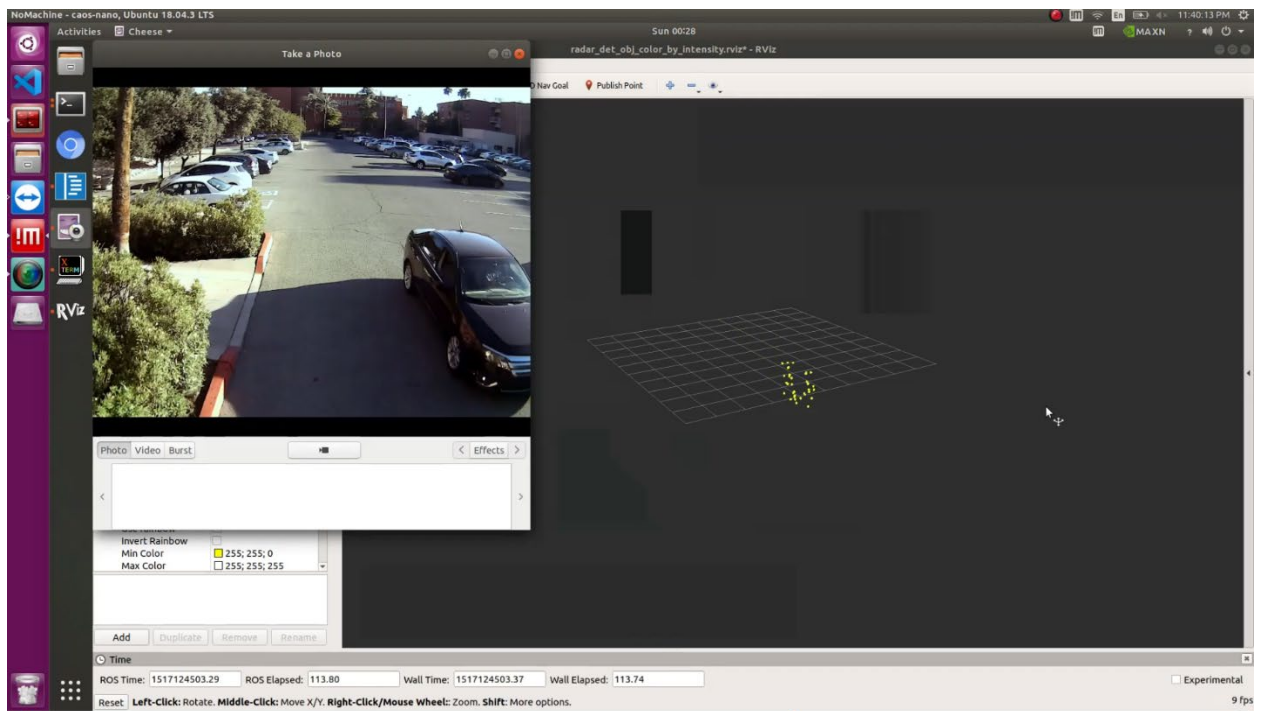
2.2 REAL-TIME DATA COLLECTION

Detailed data collection instructions are as:

1. Set up a wifi router to build a LAN.
2. Plug in the wifi module to Nvidia Jetson Nano.
3. On the first time, power up Nvidia Jetson Nano using HDMI LCD screen, keyboard and mouse. And connect to the wifi router using correct password. And then install the NoMachine software, which is for remote control, on Nvidia Jetson Nano. Set NoMachine to automatically start when the system starts. Note that, this step is only once.
4. Power off Nvidia Jetson Nano and remove LCD screen, keyboard and mouse, but keep the wifi module on it.
5. Connect the TI AWR1843BOOST radar sensor and camera to the Nvidia Jetson Nano through USB cable.
6. Power on Nvidia Jetson Nano. Then it will automatically connect to the wifi router using stored password.
7. Power on radar sensor.
8. On laptop side, install NoMachine software and connect to the same wifi router.
9. On laptop side, open NoMachine and search for the Nvidia Jetson Nano which is in the same LAN. Connect to Jetson Nano through NoMachine, and it will show a small screen of Jetson Nano.
10. Now, you can control Jetson Nano remotely.
11. To open the radar, execute the commands:
 - a. `sudo chmod 666 /dev/ttyACM0`
 - b. `sudo chmod 666 /dev/ttyACM1`
 - c. `roslaunch traffic_monitoring traffic_monitoring.launch`
12. To record the radar data on USB stick, you first need to insert the USB stick in Jetson Nano. And switch to USB stick folder. And then execute the command:
`roslaunch traffic_monitoring/traffic_monitoring/radar_scan /traffic_monitoring/radar_scan_markers /usb_webcam/image_raw/compressed -b -0 -O file_name.bag`



(a) Previous frame



(b) Later frame

Figure 2.2: Real-time data collection for a running sedan. (a) Left is video view; (b) Right is radar view

For the training data collection, because the GMM fitting is an unsupervised way, we kept one person continuously walking in the radar detection area, and one car driving through periodically. For the testing data collection, because the ground truth is needed to evaluate the model performance, we let the person walk on the left side of the radar line-of-sight ($y = 0$), and the car drive on the right side. Then we labeled all the points with centroid ($y > 0$) as a pedestrian, all the points with centroid ($y \leq 0$) as a car, and all the points without an associated centroid as clutter.

Finally, we collected 8,035 frames of training data with a duration of about 13 minutes, and 1,222 frames of testing data with a duration of about two minutes.

2.3 EXPERIMENT RESULTS

We used the scikit-learn APIs to fit the GMM using the training dataset, and saved the model to disk. Then we used the saved model to predict the testing dataset. Because the GMM fitting is an unsupervised approach, the GMM does not necessarily predict the same label as the ground truth. For example, the GMM may predict the pedestrian as an integer label, say, 0, while the ground truth for pedestrian would be another integer label, say, 1.

So, we visually associated the prediction label with the ground truth label. Fortunately, for a saved GMM model, this manual association just needs to be quickly done once.

Finally, we evaluated the segmentation results. Fig. 2.3 shows one frame of the results. Referring to Fig. 1.3, it is one example before the segmentation as all the radar points are colored in black, which means it has no class information.

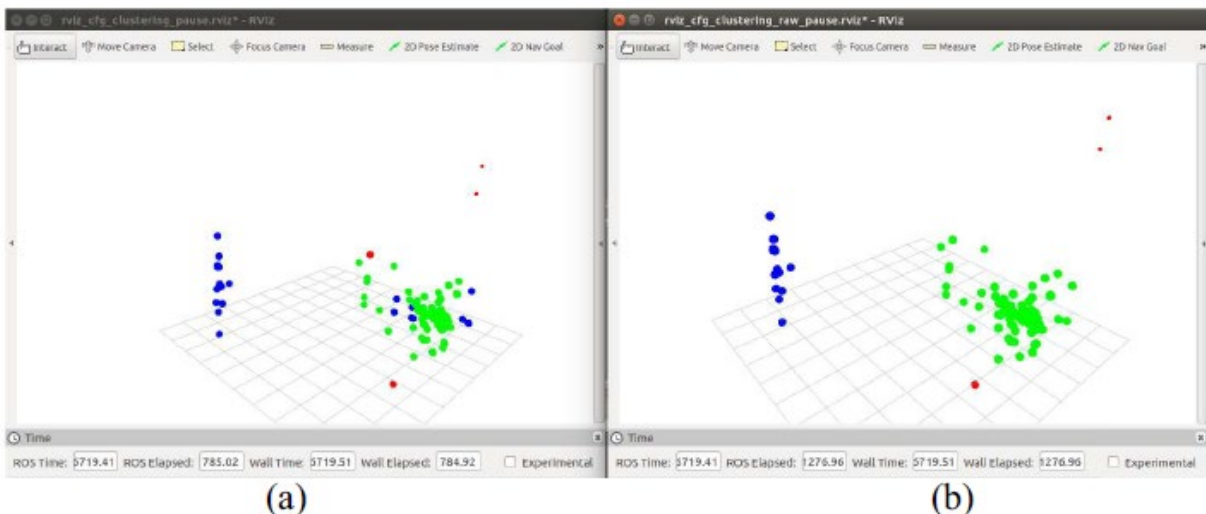


Figure 2.3: One frame of results. Red point: clutter; Green point: car; Blue point: pedestrian. (a) Prediction result shows that for the pedestrian on the left most of its points are correctly classified as blue, which is very promising, while for the vehicle on the right there are a few clutter points as in red. (b) Ground truth.

To evaluate GMM in radar point cloud segmentation, we calculated the precision, recall and intersection-over-union (IoU) [19] as the performance metrics, as in the traditional image/LiDAR segmentation domain. In the interpretation of these metrics, the precision is intuitively the confidence that the model correctly classifies a point, and the recall is intuitively the confidence the model does not miss the detection of this object class. From the perspective of radar signal processing, high precision means a low false alarm rate; high recall means a low missed detection rate.

Thus, a good model should have high precision and high recall simultaneously. And the F1 score, which is equal to $\frac{2}{precision^{-1}+recall^{-1}}$, can be interpreted as one value metric of this model. The IoU, also called the Jaccard index, represents the percentage of overlap between the prediction and the ground truth. According to [19], the IoU is recognized as the segmentation accuracy, and a model with 50 percent overlap is considered good by standard. The results of GMM on the radar point cloud was presented in Table II. As we can see here, the IoU of both pedestrian and car is above 50 percent.

Table 2.1: Performance matrices

	Precision	Recall	F1 Score	IoU
Clutter	0.71	0.89	0.79	0.66
Car	0.88	0.61	0.72	0.56
Pedestrian	0.85	0.93	0.89	0.80

To further evaluate the model performance, we plot the precision-recall curve as shown in Fig. 2.4. In general, a point will be classified into class A, if the posterior probability of class A is greater than the threshold = 0.5. As we adjust this threshold, the precision and recall changes accordingly.

Normally, if we increase this probability threshold the precision will be increased while the recall will be decreased and vice versa. The precision-recall curve shows the trade-off between these two. A good model has a position with both high precision and high recall. And we also computed the confusion matrix, as shown in Fig. 2.5.

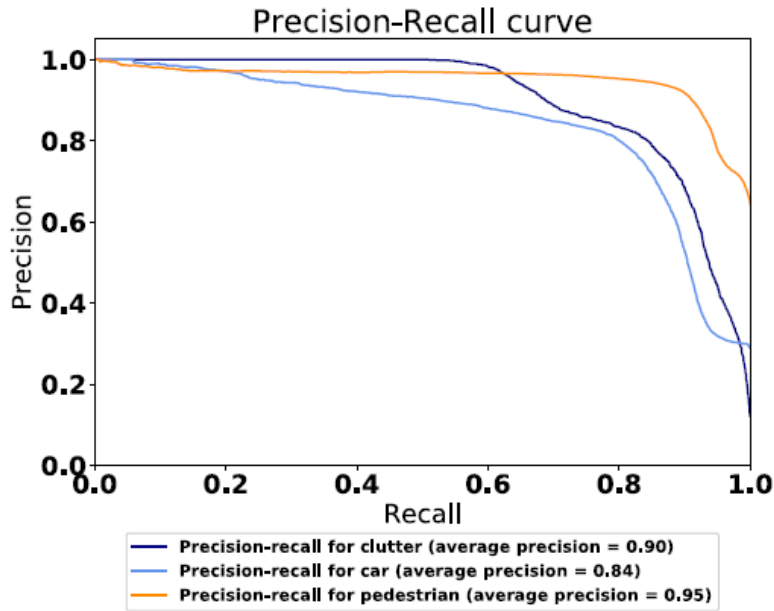


Figure 2.4: Precision-recall curve

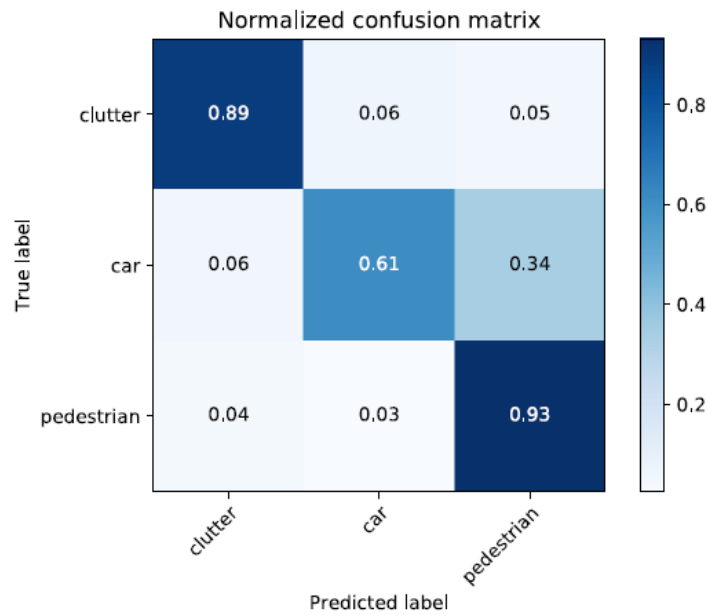


Figure 2.5: Confusion matrix

Due to some difficulties in the data collection, the collected car data was less than the pedestrian data. Thus, the GMM model fitting was biased more on the pedestrian. And the performance of car point classification was relatively poor compared to that of the pedestrian. This can be alleviated if more data for both classes can be collected.

It is noted that this result is on a single-frame basis, which means we do not accumulate multiple frames, such as in [6], as accumulation of multiple radar frames does not make sense in a scenario where vehicles are moving at high speed.

As a comparison, LiDAR is more popular in a point cloud segmentation research as LiDAR provides more points with higher angular resolutions. In recent, researchers can achieve very high performance with more object class segmentation [20-22]. The performance we achieved is relatively worse than the star-of-art LiDAR segmentation results. However, our work is very promising as radar is more suitable than LiDAR for a traffic monitoring application.

3.0 PROJECT CONCLUSION

In this study, we used a mmWave radar to capture the radar point cloud in which there are three kinds of objects (i.e., clutter, pedestrian and sedan). Then we implemented the GMM to perform the segmentation (i.e., the pointwise classification), and calculated the performance metrics such as precision, recall and IoU. And we found the GMM is simple but effectively achieves promising segmentation results.

In the future, we aim to put the device at a traffic intersection to continuously collect more data with more transportation modes, such as pedestrian, motorcycle, bicycle, sedan, truck and bus, to further evaluate the GMM performance. As we expect, as the data complexity is increased, the simple GMM would fail to achieve a good performance.

However, we will use the GMM as a preliminary classifier to help the DBSCAN algorithm, whose parameters are object specific, to more robustly group the radar points from one object as one cluster. In return, the correctly clustered points will improve the object classification accuracy. So, the work in this report will be a part of our future work, which is to implement a joint clustering/tracking and classification in the multimodal traffic monitoring application using the mmWave radar sensor. In the meanwhile, we will explore other detection devices which are also robust to adverse weather conditions and poor illumination, and then we will build a fusion framework that take advantage of multiple detection methodologies.

4.0 REFERENCES

- [1] “AWR1843 single-chip 76-GHz to 81-GHz automotive radar sensor integrating dsp, mcu and radar accelerator,” Texas Instruments. [Online]. Available: <http://www.ti.com/product/AWR1843> (Accessed on Nov. 18, 2019).
- [2] “TEF810X fully-integrated 77 GHz radar transceiver,” NXP Semiconductor. [Online]. Available: <https://www.nxp.com/products/rf/radar-transceivers/tef810x-fully-integrated-77-ghzradar-transceiver:TEF810X> (Accessed on Nov. 18, 2019).
- [3] “Radar sensors for automotive,” Infineon Technologies. [Online]. Available: <https://www.infineon.com/cms/en/product/sensor/radarimage-sensors/radar-sensors/radar-sensors-for-automotive/> (Accessed on Nov. 18, 2019).
- [4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.
- [5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.
- [6] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, “Semantic segmentation on radar point clouds,” in 2018 21st International Conference on Information Fusion, July 2018, pp. 2179–2186.
- [7] N. R. Pal and S. K. Pal, “A review on image segmentation techniques,” *Pattern Recognition*, vol. 26, no. 9, pp. 1277 – 1294, 1993. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/003132039390135J>
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. NY, USA: Springer, 2016, ch. 9.1.
- [9] C. Aggarwal and C. Reddy, *Data Clustering: Algorithms and Applications*. CRC Press, 2018.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.
- [11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems 30*, I. Guyon et al., Eds. Curran Associates, Inc., 2017, pp. 5099–5108.

[Online]. Available: <http://papers.nips.cc/paper/7095-pointnet-deephierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf>

[12] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[13] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. NJ, USA: Prentice-Hall, Inc., 2009, ch. 5.

[14] E. Rignot and R. Chellappa, "Segmentation of polarimetric synthetic aperture radar data," *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 281–300, July 1992.

[15] O. Schumann, C. Wöhler, M. Hahn, and J. Dickmann, "Comparison of random forest and long short-term memory network performances in classification tasks using radar," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Oct 2017, pp. 1–6.

[16] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2d car detection in radar data with pointnets," *CoRR*, vol. abs/1904.08414, 2019. [Online]. Available: <http://arxiv.org/abs/1904.08414>

[17] xWR1843 Evaluation Module (xWR1843BOOST) Single-Chip mmWave Sensing Solution, Texas Instruments, Aug. 2019. [Online]. Available: <http://www.ti.com/lit/ug/spruim4a/spruim4a.pdf>

[18] "Jetson nano developer kit," Nvidia. [Online]. Available: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit> (Accessed on Nov. 18, 2019).

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010

[20] Shinohara, T.; Xiu, H.; Matsuoka, M. FWNet: Semantic Segmentation for Full-Waveform LiDAR Data Using Deep Learning. *Sensors* 2020, 20, 3568. <https://doi.org/10.3390/s20123568>

[21] Zixiang Zhou, Yang Zhang, Hassan Foroosh; *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13194-13203

[22] Yiming Zhao, Xiao Zhang, Xinming Huang; *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2021, pp. 2464-2473