MmWave Radar Point Cloud Segmentation using GMM in Multimodal Traffic Monitoring

Feng Jin*, Arindam Sengupta*, Siyang Cao* and Yao-Jan Wu†
*Department of Electrical and Computer Engineering
†Department of Civil and Architectural Engineering and Mechanics
The *University of Arizona*, Tucson, AZ 85721
Email:{fengjin, sengupta, caos, yaojan}@email.arizona.edu

***Abstract*—In the 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. Thus, we propose to use a high-resolution 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. 'point-wise' 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. In the future, we aim to extend this approach to a joint clustering/tracking and classification algorithm.**

***Index Terms*—mmWave radar, radar point cloud, segmentation, Gaussian mixture model, classification, traffic monitoring.**

I. Introduction

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 3-D space is referred to as a radar point cloud, derived from a similar terminology used for LiDAR point cloud. And the radar point cloud segmentation is the point-wise classification, which means it would classify each reflection point into a single class. Segmentation for data obtained using the 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, the 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 provide the capability to offer range resolution of up to 5 cm on account of an ultra-bandwidth of up to 4 GHz, using carrier frequencies of 60GHz, 77GH and 80GHz, depending on the area of application. Furthermore, with the advanced semiconductor fabrication process, more radio frequency (RF) channels are interpreted into a single monolithic microwave integrated circuit (MMIC) chip. This allows compact mmWave radars to provide relatively better angle resolution compared to outdated bulky commercial radars. Several examples of these mmWave FMCW MMIC radar chip 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 yields very promising results.

Particularly, in the 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 paper, we use one high-resolution mmWave radar device to monitor an experimental scene with pedestrian and car in it, and gather the radar point clouds. We propose to compute a new feature vector for each radar point. Then we use the multivariate Gaussian mixture model (GMM) as the decision algorithm to perform the radar point cloud segmentation, i.e. point-wise classification. The structure of this paper is as follows: Section II presents a review of the current segmentation techniques; Section III summarizes the multivariate Gaussian mixture model and the radar point feature vector we obtain from the mmWave radar point cloud data; Section IV presents the experimental setup and validation results; and Section V concludes this paper and proposes a future work.

II. Literature Review of Segmentation Techniques

This section reviews some latest segmentation techniques in the application domains of image, LiDAR and radar processing, as shown in Fig. 1.

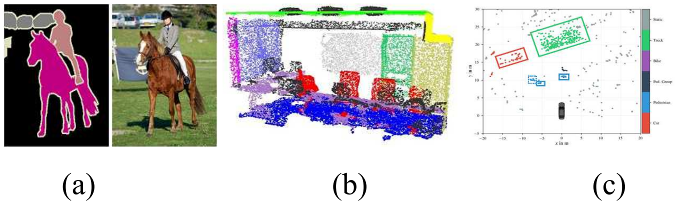

(a) (b) (c)

Fig. 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].

A. Image Segmentation

Although the image segmentation has very broad approaches with a long researching 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, while the radar cross-section (RCS), i.e. the similar term as the intensity in the radar point cloud, 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), may build a model to estimate the density or intensity-scale of the pixels, would be considered into radar point cloud segmentation. Specifically, the k-means is to assign 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 with the distances to points outside of the cluster [8]. The Gaussian mixture model is to model a group of data as a weighted sum of Gaussian distributions where each one accounts for one kind of cluster, and a cluster is formed if all the points obey the same Gaussian distribution [9]. In the DBSCAN, a core point is such that if its neighborhood of a given radius contains at least a given minimum number of points. Then, the DBSCAN algorithm is to form a cluster for all density-reachable points, i.e. each point is within the neighborhood of the core point, and all density-connected points, i.e. there is a third point from which both of these two points are reachable [9].

Recently, the deep learning based approaches achieve very promising results in the image segmentation. In [4], the authors proposed a fully convolutional network (FCN) 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 ROI along with CNN features computation, and then to classify the region using a linear support vector machine (SVM). The success of supervised deep learning approaches motivates the researchers to apply it on the LiDAR point cloud segmentation.

B. LiDAR Point Cloud Segmentation

Each LiDAR point contains the information of 3D position and intensity. With the 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 from the object, which is a pioneering work on LiDAR point cloud directly processing, compared with the traditional ways may have voxelization first which is 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.

And in the PointNet++, as 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 multi-scale and multi-resolution grouping to deal with the variation in different areas, (iii) and the farthest point sampling (FPS) to sample the points in a more efficient way.

Finally, in the 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 following by a PointNet++ model to do the classification. The PointNet family can do one object classification and scene segmentation.

C. Radar Point Cloud Segmentation

Although the segmentation has been applied in the synthetic aperture radar (SAR) image processing [14] many years ago, the segmentation on the radar point cloud has a very short history. This is because the previous real aperture radar has a very limited resolution so that the segmentation may not have significant results, while the SAR has a relatively better resolution.

With the great success of the PointNet family on LiDAR or 3D point cloud processing, researchers want to try it out on the radar point cloud and expect promising results as well. In [6], the authors first accumulated multiple radar frames to get richer point cloud, then applied the Frustum PointNet with some minor adaptations on the 2D radar point cloud, and claimed a better segmentation results over their previous work [15] in which they used a combination of the DBSCAN and long short-term memory (LSTM) network to predict the class for each radar point. And in [16], the authors applied the PointNets on the 2D radar point cloud to differentiate the vehicle from clutters with the vehicle bounding box estimation.

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

III. Gaussian Mixture Model and MmWave Radar Point Cloud

Among all the available segmentation techniques from different application domains, we think the GMM model along with the relatively high-resolution mmWave radar data would be a feasible way to implement the radar point cloud segmentation.

A. 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 totally K classes these points belong to. For a data point x given that it belongs to the k-th class, i.e. $c_k = 1,\ k \in \{1, ..., K\}$, it is assumed to follow a certain multivariate Gaussian distribution as

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

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, that is a linear superposition of Gaussian distributions of all the K classes, by the *law of total probability*, as

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

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. $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 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

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

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 posterior (MAP) criterion to determine the class of each radar point, that is

$$k = \max_{j} \gamma(c_j), \quad j \in \{1, ...K\}, \tag{4}$$

Thus, the remaining question is how to determine all the parameters in GMM. The expectation-maximization (EM) algorithm can be applied as followings. Assuming a set of data points $X\{x_1, ..., x_N\}$ with unknown classes is collected, the optimal parameters $(\boldsymbol{\pi}^o, \boldsymbol{\mu}^o, \boldsymbol{\Sigma}^o)$ are those to maximize the likelihood function based on the maximum likelihood estimation (MLE), that is

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

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

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

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

$$\pi_k^o = \frac{N_k}{N}, \tag{8}$$

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. 2, will iteratively update the parameters until the convergence of either the parameters or the log likelihood has been achieved.

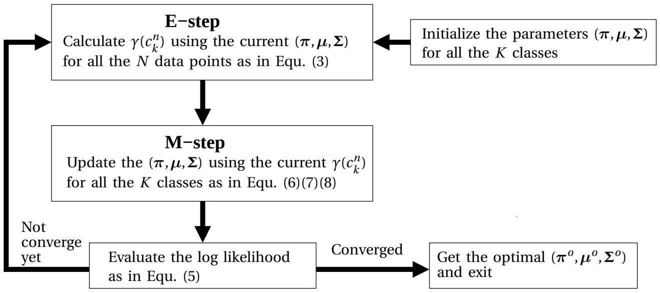

E−step
Calculate $\gamma(c_k^n)$ using the current $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ for all the N data points as in Equ. (3)
Initialize the parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ for all the K classes
M−step
Update the $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using the current $\gamma(c_k^n)$ for all the K classes as in Equ. (6)(7)(8)
Not converge yet
Evaluate the log likelihood as in Equ. (5)
Converged
Get the optimal $(\boldsymbol{\pi}^o, \boldsymbol{\mu}^o, \boldsymbol{\Sigma}^o)$ and exit

Fig. 2. The EM algorithm.

B. 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 will get the radar point cloud, in which each point is a vector of $x(r, \theta_{az}, \theta_{el}, vD, snr, noise, pX, pY, pZ, vX, vY, vZ)$. Its parameters are listed in Table I, 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 I
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

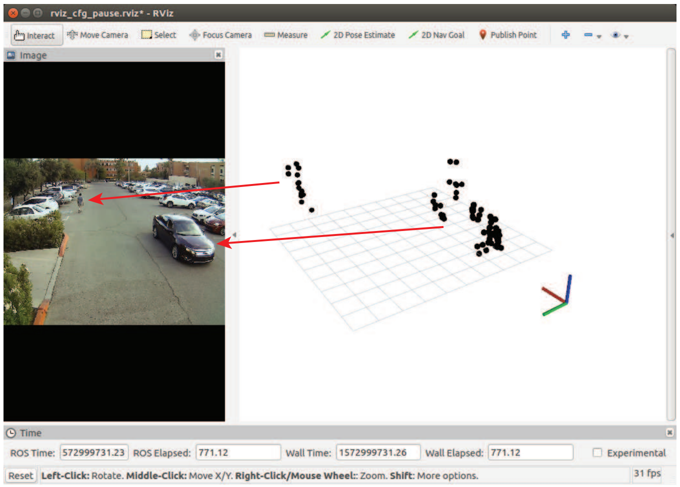
$$\Delta x = r * cos(\theta_{el}) * sin(\theta_{az}) - pX, \tag{9}$$

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

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

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

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



ROS Time: 572999731.23 ROS Elapsed: 771.12 Wall Time: 1572999731.26 Wall Elapsed: 771.12 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options. 31 fps

Fig. 3. MmWave radar point cloud example. Red axis: x; Green axis: y; Blue axis: z.

As an 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 (RCS) in the unit of dBsm. We view the generation of 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 distinguishable than these of a sedan, so the GMM will can be applied for classification between these two. It is the same for the other transportation modes. Fig. 3 shows an example of radar point cloud including a car and a pedestrian from our data collection that will be described in Section IV. Here we can see the differences between the distributions of points from these two kind of objects.

IV. EXPERIMENT SETUP AND FIELD TEST IN MULTIMODAL TRAFFIC MONITORING

A. Experiment Setup and Data Collection

We used the TI mmWave radar 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. 4 shows how we collected the experimental data in a parking lot. The device was raised up to 3 meters high, and all the data was wirelessly transferred to a laptop for storage.

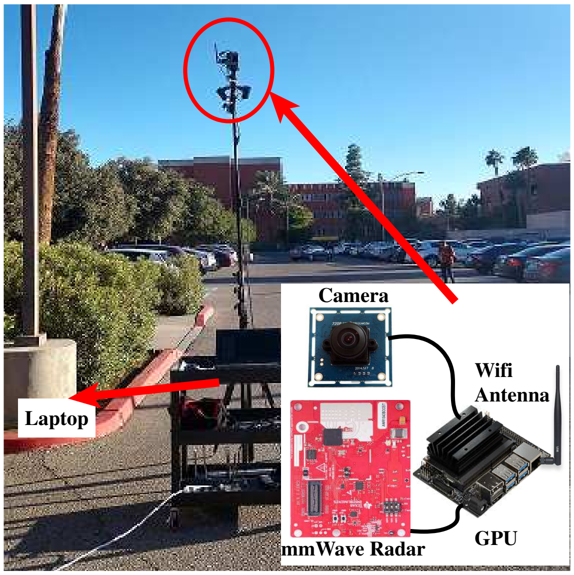


Fig. 4. Experimental setup.

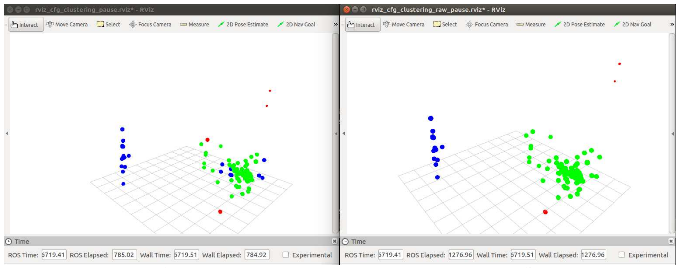
With the proper FMCW waveform design and the implementation of multiple-input-multiple-output (MIMO) direction-of-arrive (DOA) algorithm, we achieved about 0.09 meters of range resolution, 0.8 m/s of Doppler resolution, 15 degrees of azimuth angle resolution, and 28 degrees of elevation angle resolution. 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.

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 frequently. For the testing data collection, because the ground truth is needed to evaluate the model performance, so we let the person walking on the left side of the radar line-of-sight ($y = 0$), and the car driving on the right side. Then we labeled all the points with centroid ($y > 0$) as a pedestrian, all the points with centroid ($y <= 0$) as a car, and all the points without associated centroid as clutter.

Finally, we have collected 8035 frames of training data with a duration of about 13 minutes, and 1222 frames of testing data with a duration of about 2 minutes.

B. Experimental Results

We used the scikit-learn APIs to fit the GMM using the training dataset, and saved the model in the 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 number, saying, 0, while the ground truth for pedestrian would be any other number, saying, 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 got the correct segmentation results. Fig. 5 shows one frame of the results. Referring to Fig. 3, it is one example before the segmentation as all the radar points are colored in black, which means it has no class information.



ROS Time: 5719.41 ROS Elapsed: 785.02 Wall Time: 5719.51 Wall Elapsed: 784.92

ROS Time: 5719.41 ROS Elapsed: 1276.96 Wall Time: 5719.51 Wall Elapsed: 1276.96

(a) (b)

Fig. 5. One frame of results. Red point: clutter; Green point: car; Blue point: pedestrian. (a) Prediction. (b) Ground truth.

To evaluate the performance of 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 exceeding 50% overlap can be considered as a good one. 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%.

TABLE II
PERFORMANCE METRICS

	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 plotted the precision-recall curve as in Fig. 6. In general, a point will be classified into class A, if the posterior probability of class A is greater than the $threshold = 0.5$. If we adjust this threshold, the precision and recall will be changed. Normally, if we increase the probability threshold, the precision will be increased while the recall will be decreased; 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. 7.

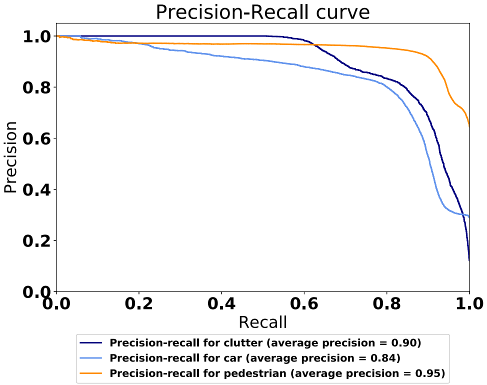


Fig. 6. Precision-recall curve.

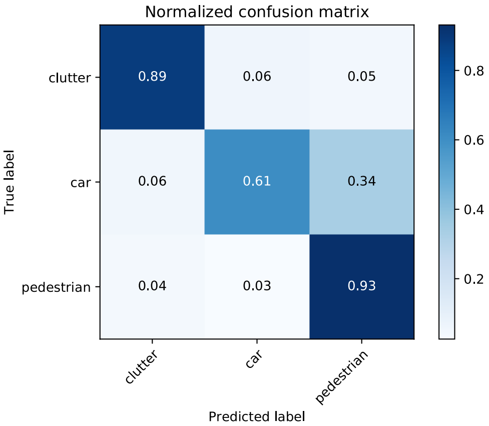


Fig. 7. Confusion matrix.

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 than 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], that does not make sense for a high-speed vehicle.

V. Conclusion and Future Work

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 do the segmentation, i.e. the point-wise 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 will 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 paper 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.

Acknowledgement

This project was funded by the National Institute for Transportation and Communities (NITC; grant number 1296), a U.S. DOT University Transportation Center, and also supported by City of Tucson.

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-ghz-radar-transceiver:TEF810X (Accessed on Nov. 18, 2019).

[3] "Radar sensors for automotive," Infineon Technologies. [Online]. Available: https://www.infineon.com/cms/en/product/sensor/radar-image-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-deep-hierarchical-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.