LiDAR-based Vehicle Detection by using DBSCAN Unsupervised Clustering approach

1st Tri Bien Minh
Electrical and Computer Engineering
Vietnamese German University
Binh Duong, VietNam
tri.bm@vgu.edu.vn

2nd Hien Vo Bich
Electrical and Computer Engineering
Vietnamese German University
Binh Duong, VietNam
hien.vb@vgu.edu.vn

***Abstract*— Roadside LiDAR is helping to build intelligent and safe transportation. Object detection is a challenging and fundamental problem in computer vision. Moreover, the vehicle detection system is essential to Intelligent Transportation Systems (ITS). Many researchers in the transportation field spend an enormous amount of money to collect and analyze traffic data to optimize street systems. This research aims to develop a case study for a vehicle detection system in a complex roadway area based on LiDAR through an embedded system. For this purpose, an embedded GPU integrated (Nvidia JetsonTX2) with low power and high performance has been picked, which supports an unsupervised learning algorithm to be run simultaneously and a detection algorithm to be applied for point cloud recognition. We also discuss the structure of the architectures of LiDAR-based roadside systems, and lidar data processing for vehicle detection. In the real-scanned HDL-32E Velodyne 3D LiDAR dataset, our proposed method can achieve vehicle detection accuracy up to 82.7% in several real-scenes datasets. The future research directions to contribute resources beneficially to industry, academia, and government agencies for choosing appropriate LiDAR-based technologies for their vehicle monitoring systems.**

Keywords*— *LiDAR, vehicle detection, traffic monitoring, unsupervised learning, Jetson, DBSCAN, roadside lidar

I. INTRODUCTION

As the number of vehicles has increased dramatically, this causes severe traffic congestion in big cities in many countries. One of the good solutions to reduce traffic congestion is a traffic monitoring system [1]. The Intelligent Transportation System (ITS) is used to gather traffic data such as the number of vehicles, vehicle speed, and types of vehicles. These collected data can be used for traffic analysis to enhance the safety of transportation, predict future transportation demands, and take advantage of roadway systems [2]. Vehicle detection is the main functionality of the traffic monitoring system. Due to dramatic technical challenges, various research papers have been considered regarding vehicle detection systems. The contributions of this article are summarized as follows.

- This article is using the architecture with computing at the edge using a low-power embedded computer Jetson TX2.
- Propose a collecting and processing hardware and processing point cloud data pipeline.
- Experiment with vehicle detection based on DBSCAN unsupervised learning methods with different scenes.

II. ARCHITECTURE OF THE LIDAR ROADSIDE SYSTEM

The system architecture of the LiDAR-based connected roadway infrastructure integrates roadside LiDAR sensors, traffic data processing system, and software-connected road users. The system is shown in Fig.1.

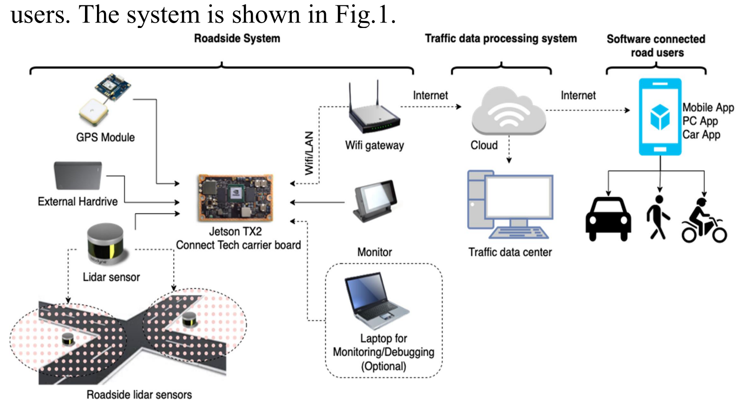

Fig. 1. The system architecture of the roadside LiDAR-based connected roadway infrastructure

A. *Roadside LiDAR sensors*

LiDAR sensors can be installed at two or four corners to cover the full coverage of the intersection area, which depends on the intersection size and geometry of the roadway. The 360-degree 3D LiDAR sensors were used for vehicle detection. In our work, the roadside LiDAR sensors used high-resolution HDL-32E and average-resolution VLP-16 LiDAR sensors from the Velodyne brand. The HDL-32E LiDAR can generate 360-degree 3D point cloud data of up to 1.39 million points per second by using 32 internal laser/detector, which detects a range of up to 100 m and an average accuracy of ±2 cm. VLP-16 LiDAR has only 16 laser/detector, which can create 600,000 points per second, and the range is also up to 100 m, which lowers the accuracy to only ±3 cm. The rotational speed of the LiDAR is about 5-20 rotations per second, which covers the vertical field of view (FOV) about 45-degree for HDL-32E

LiDAR and 30-degree for VLP-16 [3]. Maintaining the Integrity of the Specifications

B. Embedded computer

Embedded computer at each intersection is to process collecting data from LiDAR sensors. In this study, the Jetson TX2 has been the main processor unit. It has integrated the GPU with 256-CUDA cores, Quad-Core ARM Cortex A57, 8GB memory, and 32 GB disk space. Power saving is the unit key point of this module, which consumes only 7.5W~15W [4]. It is suitable for AI computing at the edge applications, especially computer vision applications.

C. External hard drives and Peripherals

Hard drives are used for storing and collecting data, which are optional and depend on the architecture of the systems. If the embedded computer can process the real-time data and send this data over the internet through the gateway, in this case the hard drives are not necessary. Otherwise, data can be stored on the hard disks for backup, diagnostic and debugging applications. The hard drives have not been used in our system, because the main approach in our system is toward computing at the edge, and uploading these data to the cloud server. Other peripherals such as laptops, monitors, light sensors, and GPS-module are helping the system work more reliably and easy to monitoring and debug systems.

D. Traffic data processing

Data collected from the roadside system have been sent to the cloud and traffic data center. The traffic data center is for data archiving, data integration, performing the traffic operation, traffic control with optimization decisions, and helping the human make the most suitable choices to control the system. This data center can also help the road users fletch, and visualize the information from the real situation in the roadway.

E. Software-connected road users

Users can get real-time road information about congestion status, numbers of vehicles on roads, the density of vehicles on the roads, average moving speed in consider areas, and related traffic information. The software can be developed on multiple platforms and devices such as smartphones, tablets, PC, and car HMI.

III. LIDAR DATA PROCESSING

The vehicle detection process and visualization LiDAR data processing steps are shown in Fig.2.

A. Point cloud Preprocessing step

LiDAR can generate a huge amount of 3D point clouds which require high computing powers. Therefore, the raw point cloud must be filtered before going to the ground point removal step. The crop point cloud is adopted to filter the sparse points within a distance greater than 50m from LiDAR, and remove the center lane area. Voxel grid filtering created a cubic grid and filtered the point cloud by only leaving a single point per voxel cube.

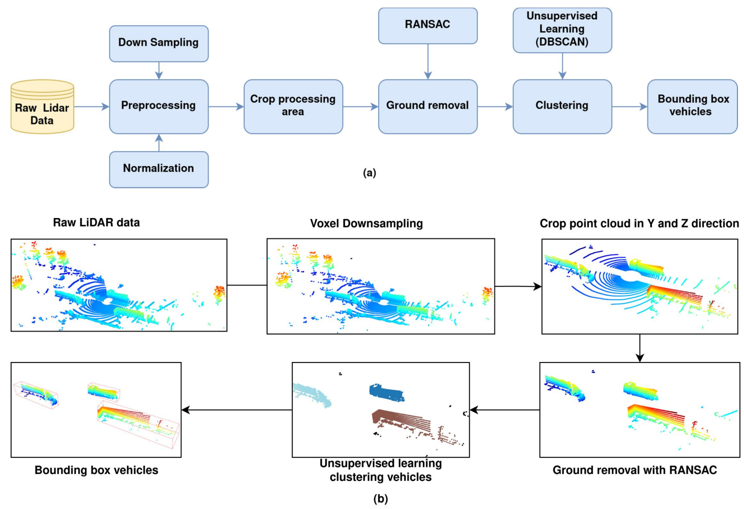

Fig. 2. Vehicles detection process (a) and visualization LiDAR data processing step (b)

Then the larger the cube length the lower the resolution of the point cloud, which is the number of points data also decreasing, which can be seen in Fig. 3.

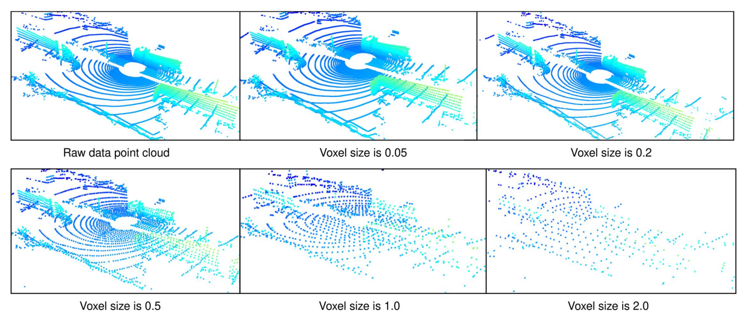

Fig. 3. Voxel down sampling with various voxel sizes between 0.05 to 2.0

B. Ground point removing

LiDAR with 45/30-degree vertical FOV can create a ground point cloud of the road when they are working. These ground points take up a large proportion of the raw point cloud, which intensely affects the subsequent processing. In this step, the RANSAC (Random Sample Consensus) method was used to filter the ground plane [6]. The RANSAC operates for a max number of iterations and returns the model with the best fit. Each iteration randomly picks a subsample of the point cloud data and fits a model through a point cloud plane. Then the iteration with the highest number of inliers or the lowest noise is used as the best model. The RANSAC algorithm for ground point removal is described below.

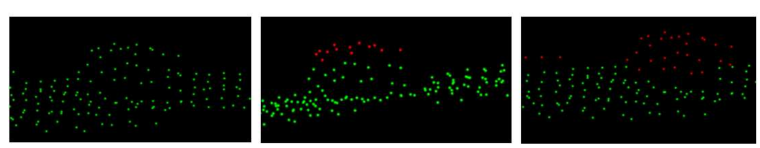

Fig. 4. Result of the RANSAC with number of iterations is remaining 100, and different threshold distance 1.5 (left), 0.7 (middle), 0.3 (right)

When changing the different threshold distances the inliers were changing, and the number of iterations can also affect the computing time and finding the best fit model. In our work, the good number for getting a good result is a threshold distance value of 0.3 and the iteration value of 100. The result of the RANSAC is shown in Fig. 4. The red point cloud is the outliers, and the green one is the inliers after running the algorithm.

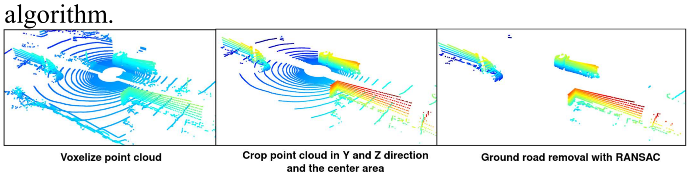

Fig. 5. Processing steps for RANSAC: Voxel down sampling to reduce point cloud data (left) Crop in the Y and Z-axis and center area (middle) Result of the RANSAC filter out the road plane (right)

Processing steps for RANSAC have followed steps, as shown in Fig. 5: Voxel down sampling to reduce point cloud data, next crop point cloud data in the Y and Z-axis and center area, then using RANSAC for plane segmentation to remove road plane.

C. DBSCAN Unsupervised clustering approach

To combine local point cloud clusters given a point cloud from LiDAR sensor. Clustering algorithms can be used for this objective. It includes a variety of techniques based on various distance units. For instance, Gaussian mixtures (Mahalanobis distance to centers) [7], Affinity propagation (graph distance) [8], Mean-shift (distance between points) [9], and Spectral clustering (graph distance) [10]. The researchers developed unsupervised learning techniques that could build feature detector layers without the need for labeled data. Deep learning interest was rekindled in part due to unsupervised learning. Unsupervised learning will likely become much more significant in the long run [11].

This study uses the Density-based spatial clustering of applications with the noise clustering method. Density-Based Clustering to the concept that a cluster in data space is a contiguous region of high point density, separated from other similar clusters by contiguous regions of low point density [12], clustering refers to unsupervised learning approaches that discover unique clusters in the data. The unsupervised learning DBSCAN algorithm for vehicle clustering is shown below.

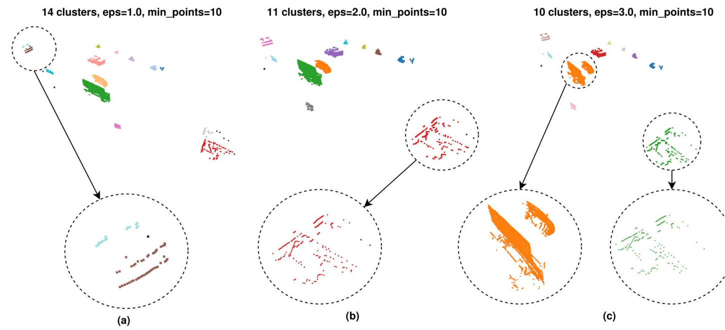

Fig. 6. Example DBSCAN with various eps value between 1.0 to 3.0 with the same minimum number of points is 10 points.

The DBSCAN is an unsupervised learning algorithm, so it does not need to train before running the clustering, this is the main advantage of this approach. But it has some drawbacks, Fig. 6 illustrates an example when changing the *eps* number and remaining the minimum point in each cluster, which has reduced the number of clusters when increasing the *eps* value. Following this experiment, DBSCAN is good for clustering the various vehicle, even small motor-bike vehicles. But it is not the optimal solution, in Fig. 6 (a) the *eps =1.0* then the neighbor in the car object is a mixture, and the algorithm returns two clusters in one object. Moreover, in Fig. 6 (b-c) the *eps =2.0 - 3.0*, the number of clusters is decreased, and get more clustering errors by grouping two objects into one cluster. It is quite tricky to choose the optimal *eps* and minimum points in this algorithm.

IV. EXPERIMENTAL RESULT

This section shows the roadside hardware setup and the result of experiments. A mounting plate for 3D LiDAR and housing for Jetson TX2 integrated Connect Tech carrier board is designed and printed by using 3D printing machine with PLA material, which is illustrated in Fig. 7.

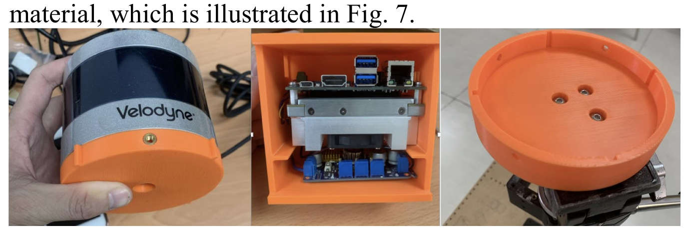

Fig. 7. 3D printing mounting object for LiDAR (left), housing Jetson TX2 (middle) and tripod mounting (right)

The roadside system of the experimental platform setting up is shown in Fig. 8. This system includes 3D Lidar HDL-32E, LCD monitoring, Ublox M8N GPS module, Jetson TX2 integrated with Connect Tech shield is the main embedded computer and 12V power supply. The system connects all the modules together and works properly in normal daylight conditions.

Fig. 8. Roadside system setup in the field test (left), setting lidar at the middle of the lane (right)

The roadside system is setup at the location, which longitude and latitude location are *[11.014811, 106.662682]*, respectively, which is shown in Fig. 9.

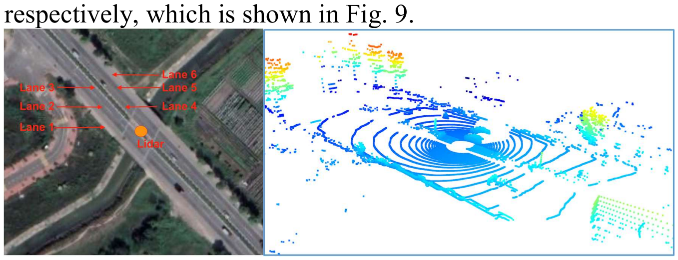

Fig. 9. Field site for collecting point cloud data in geometry location [11.014811, 106.662682] (left), sample LiDAR data frame (right)

In order to test the vehicle detection of the algorithm, we counted the number of down sampling points, crop and ground removal points, and the number of vehicle detection at each stage in multiple environments. The code was implemented using Python and the Open3D framework [13]. These data have been compared with the number of vehicles counted by humans.

Table 1. Result of vehicles detection in different scenes

Scene	Points number	Down sampling points number	Crop & Ground removal points number	Number of vehicle detection by algorithm	Number of vehicle counted by human
Scene 1	69120	28287	9201	3	4
Scene 2	69504	26677	8303	7	8
Scene 3	69120	26423	2679	8	10
Scene 4	69120	26789	1226	9	11
Scene 5	69504	26325	5053	11	12
Scene 6	69120	27643	6075	10	13
Total	**415488**	**162144**	**32537**	**48**	**58**

In the vehicle detection part, as shown in Table 1, the total of raw point cloud collecting was 415488 points, after down sampling the points number has 39.02% of the original data. After the crop and ground segmentation part, the number of points dramatically reduced by 92.16%, it only has 35537 points.

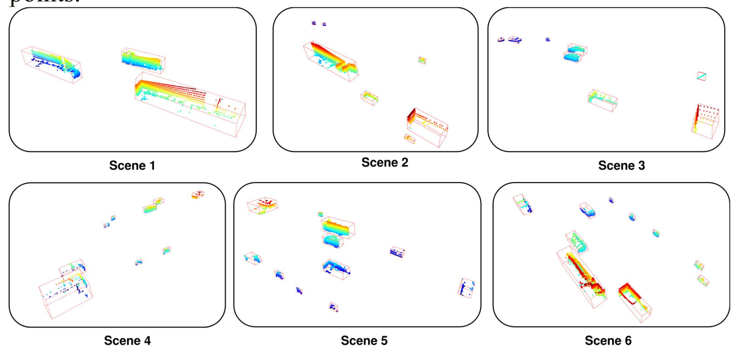

Fig. 10. Visualization of the result of vehicles detection in different scenes

The visualization of vehicle detection result in different 6 scenes, which is illustrated in Fig. 10. It is clear to see the unsupervised learning algorithm can detect mostly vehicles in the road, this can detect 48 over 58 vehicles, and vehicle detection precision was 82.7%, and some vehicles connected with bushes and cannot be separated. The four wheels vehicle detection precision was 80% over the dataset.

V. CONCLUSION

This study describes a vehicle detection procedure that mostly consists of four steps: To acquire objects above the ground by using LiDAR and get the raw point clouds, first down-sampled, crop point clouds data and ground points are eliminated, then clustering the vehicles by using the unsupervised learning DBSCAN approach. In the Velodyne HDL-32E dataset, our method achieved 82.7 % precision for total vehicle detection, and 80% precision for four wheels vehicle detection in six scenes. The error has come from the clustering process, some small vehicles such as bikes and motorbikes can be neglected, and occlusion between vehicles reduced the accuracy. Future research will take into account RGB cameras and other sensor data to increase detection accuracy. Moreover, deep learning methods are considered to deal with further detection and classification applications.

ACKNOWLEDGMENT

The authors wish to thank Mr. Linh V. Ngo for collecting data and field tests, the BugCar team in the VGU robotics laboratory for designing the 3D mounting housing, as well as providing us with the 3D Velodyne Lidar HDL-32E in a short time to run the application.

REFERENCES

[1] M. Won, T. Park and S. H. Son, "Toward Mitigating Phantom Jam Using Vehicle-to-Vehicle Communication," *IEEE Transactions on Intelligent*

Transportation Systems, vol. vol. 18, no. no. 5, pp. pp. 1313-1324, (2017).

[2] Wang, Zhao & Wang, Xing & Fang, Bin & Yu, Kun & Ma, Jie, "Vehicle detection based on point cloud intensity and distance clustering," *Journal of Physics: Conference Series,* vol. 1748. 042053., (2021).

[3] Velodyne Lidar, "VLP-16 Lidar, HDL-32E," [Online]. Available: https://velodynelidar.com.

[4] Nvidia, "Nvidia Homepage,", [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2.

[5] Bin Lv, e. al, "LiDAR-Enhanced Connected Infrastructures Sensing and Broadcasting High-Resolution Traffic Information Serving Smart Cities," *IEEE Access,* vol. vol. 7, pp. pp. 79895-79907, (2019).

[6] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* vol. 24, no. 6, p. 381–395, (1981).

[7] P. Mahalanobis, "On the generalized distance in statistics," *Proc. Nat. Inst. Sci.,* p. 49–55, (1936).

[8] Brendan J. Frey, Delbert Dueck, "Clustering by Passing Messages Between Data Points," *SCIENCE,* vol. 315, no. 5814, pp. 972-976.

[9] Cheng, Yizong , "Mean Shift, Mode Seeking, and Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* p. 790–799, (1995).

[10] Ulrike von Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing,* (2007).

[11] LeCun, Yann & Bengio, Y. & Hinton, Geoffrey, "Deep Learning," *Nature,* vol. 521, no. 436-44, (2015).

[12] Martin Ester, e. al "A density-based algorithm for discovering clusters in large spatial databases with noise," *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining,* p. 226–231, (1996).

[13] Qian-Yi Zhou and Jaesik Park and Vladlen Koltun , "Open3D: A Modern Library for 3D Data Processing," *ArXiv,* vol. abs/1801.09847, (2018).

[14] M. Won, "Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey," *IEEE Access,* vol. vol. 8, pp. pp. 73340-73358, (2020).