ASEAN Journal of Engineering and Technology, 2 (3): 01-10, 2022
e-ISSN: 2600 - 9056

DOI: 10.26666/rmp.ajte.2022.3.1

Hardware Development of Autonomous Mobile Robot Based On Actuating Lidar

Muhammad Rabani Mohd Romlay, Azhar Mohd Ibrahim, Siti Fauziah Toha*, Muhammad Mahbubur Rashid & Muhammad Syahmi Ahmad

Department of Mechatronics Engineering, International Islamic University Malaysia,53100, Gombak, Kuala Lumpur, Malaysia
*Corresponding Author: tsfauziah@iium.edu.my

Received: 05 October 2022; Revised: 15 October 2022; Accepted: 10 November 2022; Published: 20 December 2022

Abstract: Object detection using a LiDAR sensor provides high accuracy of depth estimation and distance measurement. It is reliable and would not be affected by light intensity. However, high-end LiDAR sensors are high in cost and require high computational costs. In some applications such as navigation for blind people, sparse LiDAR point cloud are more applicable as they can be quickly generated and processed. As opposed to a point cloud generated from high-end LiDAR sensors where many algorithms have been developed for object detection, sparse LiDAR point clouds still possess large room for improvement. In this research, we present the construction of an autonomous mobile robot based on a single actuating LiDAR sensor, with human subjects as the main element to be detected. From here, the extracted values are implied on k-NN, Decision Tree and CNN training algorithm. The final result shows promising potential with 91% prediction when implemented on the Decision Tree algorithm based on our proposed system of a single actuating LiDAR sensor.

Keywords: *Obstacle avoidance, object recognition, autonomous mobile robot, decision tree.*

1. Introduction

Object detection is an integral part of robotics with various applications such as autonomous cars, navigation aids or object tracking [1], [2]. There are many types of sensors which have been implemented to achieve this purpose including cameras, Inertial Measurement Units (IMU), Light Detection and Ranging (LiDAR), and ultrasonic or sonar sensors [3]–[5].

Integrating two or more types of sensor to achieve object recognition serve to be a promising prospect as the system can integrate different types of data to interpret the complex perception of the surrounding [6]. Attempts to integrate two or more sensors have been done as researchers went on to the objectives of both detecting and recognizing obstacles within the area [7]–[9]. Autonomous cars, for instance, have been equipped with LiDAR sensors as well as a camera for navigation purposes [10]. Camera and LiDAR data can be merged to achieve object recognition of road lanes [11]. DeepLiDAR [9] method, fuses LiDAR data with an image for depth prediction in outdoor space.

Few sensors of the same type can be calibrated to provide better data fusion. Multiple LiDAR sensors could provide more complete 3D-projected scanning of the surroundings as practised by Pereira et al. [6]. They used

Corresponding Author: Siti Fauziah Toha, International Islamic University Malaysia (IIUM), Jalan Gombak, 53100, Kuala Lumpur, Malaysia. Email: tsfauziah@iium.edu.my

multiple LiDARs, which are calibrated on an autonomous vehicle. These sensors can be placed on both rear sides of the vehicle, which will enable the sensors to shoot from a wider angle of view. Thus, this method helps the system to recognize objects more decisively and efficiently especially, when applied to mobile robots which have been a famous subject amongst researchers today [12].

Nevertheless, in a certain scenario having two or more types of sensors or even more sensors of the same type is less preferable, as it will increase computational cost, and power consumption and reduce mobility and portability. The calibration of the data is a concern as it would be difficult to cater to many sensors at the same time [6].

For achieving object recognition using a single unit sensor, LiDAR proved to be a suitable option. The sensor possesses high accuracy of distance measurement and as opposed to the camera, it does not depend on the light intensity of the surrounding [3]. Its detection range is also comparatively more accurate and reliable when compared to stereo methods [9]. Usually associated with high-end sensors, at times it is claimed to be high in cost [2] for consumers.

Nonetheless, high-end LiDAR does not apply to all situations. There are cases where a simple or lower-end sensor suits the purpose of the system. For example in the development of robotic navigation aid for visually impaired people, the mobile robot or smart cane developed are required to be highly compact and lightweight [13].

Currently, there are numerous object recognition algorithms to detect surrounding objects such as buildings, pedestrians and vehicles [14]–[21]. Despite this, many researchers are still bounded to large hardware requirements and high-end sensors.

To cater for the necessity and requirements of individuals such as visually impaired people as stated previously, a dependable classifier method which can work with sparse LiDAR point cloud data is required. Our research proposed the use of a single-unit LiDAR sensor for object detection and recognition. To achieve this, we developed Clustered Extraction (CE) method for human detection on sparse LiDAR point cloud data. Then, we train the extracted data with k-Nearest Neighbor (kNN), Decision Tree (DT) and Convolutional Neural Network (CNN). We conceive with the proposed state-of-the-art method; a single unit LiDAR sensor is adequate to achieve object recognition.

To summarize, our main contributions are

- Genuine data of 100 scans with 500,000 sparse LiDAR point clouds with the human subjects within varying/different backgrounds surrounding
- Hardware construction proposed consisting of a single actuating LiDAR sensor as the detection system.

2. Related Works

Generally, the types of classification of LiDAR point cloud data can be separated into two, namely end-to-end learning and non-end-to-end learning [22]. End-to-end learning refers to the classification which directly takes raw lidar point cloud data as its input. It customarily requires a larger amount of data for training and thus requires a higher computational cost [22]. The example of the end-to-end classification includes PointNet [23] and PointNet++ [24].

Non-end-to-end classification, on the other hand filters the raw point cloud data by only extracting meaningful features from it, arranging it into a set of input data and then finally implies said data on the neural network or deep learning. Example of lidar classifier which belongs to the non-end-to-end type is VoxNet [25]and SEGCloud [26].

Obstacle recognition using a single LiDAR sensor usually implies high-end and costly equipment. An attempt at end-to-end classifier extraction from RMB is proposed in [27]. The author developed a complete pipeline to distinguish outdoor 3-D urban objects. It segments the input point cloud into 4 stages of the region, from the ground, followed by low foreground, high foreground and sparse area. The object detection task was done using a rotating multi-beam (RMB), which can be a tough challenge because of the low and strongly inhomogeneous measurement density [27], [28] instigate the study of pitch angle effects in rotating multibeam measurement to look for kinematic configuration for full sphere field of view. However, both methods are done on high-end and bulkier Velodyne sensors.

SARPNET [29] detects objects through shape attention regional proposal network, while the author of [30] proposes road surface detection through terrestrial LiDAR data. [31] performs ground filtering of roadside LiDAR using a channel-based filtering algorithm. Another researcher implements roadside LiDAR data integration with revolution and rotation-based methods [32].

Compared to the classification algorithm which works with high-end LiDAR or fusion of LiDAR with other sensors, the classification method for sparse point cloud data can be said to be imperceptible. Muresan et. al [33] proposed a real-time object detection using sparse 4-layer LiDAR. The method combines an elevation grid and polar histogram to detect the obstacle. However, the method has not been compared with other benchmark sparse detection algorithms. Another sparse LiDar point cloud classifier uses 3DCNN [34] aimed to be affordable for the consumer's vehicle. The author proposed a novel voxel representation which can be calculated from a sparse point cloud. Nevertheless, the utilization of multi-frame information has yet to be proven. Xie et al. [35] suggested classification base and joint sparse representation in kernel space. Ground, buildings and trees are among the classified elements in their extensive work.

Table 1 shows some selected methods of LiDAR-based detection methods with different detection objectives and varying resolutions of LiDAR data. High-resolution datasets are obtained through merging multiple sensors, using a higher quality sensor or acquiring readily available KITTI dataset benchmark. In comparison to the high-resolution point cloud, research work on sparse LiDAR data still possesses huge room for improvement and exploration.

Therefore, in this research, we proposed a novel, state-of-the-art non-end-to-end classification of sparse LiDAR point cloud data. The proposed method is tested with a genuine point cloud for evaluation. Instead of extracting geometry features such as lines or edges, for example, we extracted the number of elements of segregated point clouds with a predetermined interval, thus forming an isolated cluster of point clouds in each x, y and z coordinate. A thorough explanation of said method will be discussed in the following part.

Method	**Dataset**	**Detected object**	**Remark**	**Ref**
3DCNN	Sparse	Pedestrian	Have not been tested on multi-frame information	[34]
Novel object detection pipeline	Sparse	Vehicle	Room for improvement robustness of object tracking	[33]
Revolution-rotation	High Resolution	Roadside elements	Difficult for integration	[32]

based method	(Multiple LiDAR sensors)			
SARPNET	High Resolution (KITTI Dataset)	Car/ cyclist	Inconsistent pedestrian detection	[29]
Novel two-layer grid structure	Decent resolution (Velodyne HDL-64)	Vehicle/ pedestrian	Utilize both deep learning-based object and contextual scene analysis	[27]

Table 1. Table of comparison for LiDAR-based object detection method.

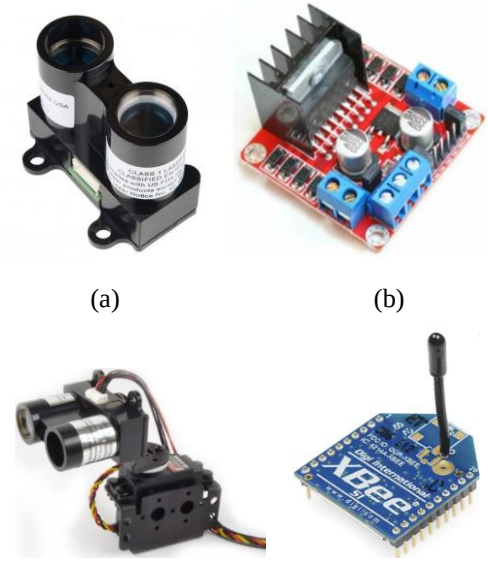
3. Construction of Hardware

In this experiment, we have selected is Garmin LiDAR Lite V3 as the detection sensor, with a distance measurement capacity of 40 meters and resolution of 1cm within +/-2.5cm accuracy. Two FS5109M servo motors with 10kg.cm torque capacity is connected via 2 brackets in order to enable the lidar sensor to rotate on the horizontal and vertical axis. A scanning degree of at least 130° is required to resemble a human's point of view [36]. Arduino Uno is used for the microcontroller (Arduino Nano can also be used for its replacement). L298N motor driver is selected to control the two tires of the mobile robot.

The system is made capable of wireless communication, via Arduino XBee embedded RF module. Through RF communication, data transfer up to 30 meters range within the indoor environment and 90 meters of outdoor line-of-sight is achieved. The RF data rate is 250,000 bps, enough for our LiDAR sensor data transmission. XBee RF module enables real-time wireless updates for control and monitoring purposes [37]. The list of items used is shown below:

- **Lidar Lite V3**
- **Arduino Nano/Uno**
- **FS5109 Servo Motor x 2**
- **Acrylic sheet frame**
- **LiPo Rechargeable Battery 2200mah**
- **Servo Bracket**
- **L298N Motor Driver**
- **Capacitor**
- **Tyre x 2**
- **Connecting wires**
- **Arduino XBee**
- **DC motor x 2**

Figure 1 below shows the figures of important hardware items used during the prototype development.



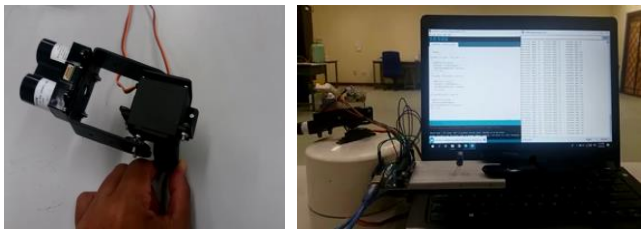
(a) (b)

(c) (d)

Fig 1. Items used include (a) LiDAR Lite V3 sensors, (b) L298N Motor Driver (c) actuating LiDAR with 2 servo motors and (d) XBee Module.

The system was first tested without implementing it on the mobile robot. The LiDAR sensor is attached to servo sensors, with 7 foundation features extracted from the LiDAR sensor. Object angle, x-axis reading, y-axis reading, z-axis reading, distance, yaw angle and pitch angle are the characteristics obtained from the scanning LiDAR. From here, other features are derived to finally achieve object recognition.

Features extraction is done on Arduino Integrated Development Environment (IDE) connected to the CPU. The CPU specifications include an Intel (R) Core (TM) i7-5500U CPU @ 2.40GHz, 8Gb RAM and a 64-bit operating system. The systems are tested within both an indoor environment (laboratory) and an outdoor environment. Figure 2 shows the detection systems which include the LiDAR sensor, servo motor, Arduino and processing unit. The detection system is initially tested with i2c communication connecting the wire to the CPU.



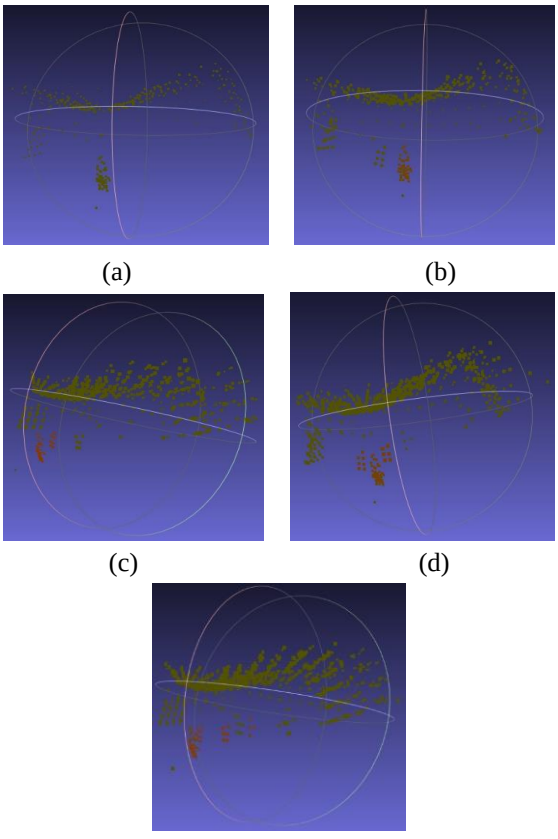
(a) (b)

Fig 2. LiDAR sensor calibration with servo motors. Fig 2 (a) LiDAR sensor with 2 servo motors & Fig (b) LiDAR sensors with Arduino for features extraction.

Following the initial test, the detection system is then attached to the mobile robot as shown in Figure 3 below. Two tires with additional DC motors are connected to the detection system. A supporting third wheel is needed for stability. With the mobile robot platform, the Li-Po battery is necessary for the external power supply. XBee RF module is attached for wireless data transmission with a detection range of over 30 meters communication distance.

Fig 3. First mobile robot prototype with LiDAR sensor for object recognition.

The system is tested for human scanning, as the human subject moves from 1 meter distance towards 5 meters distance. The visualization of LiDAR point cloud data is displayed via MeshLab software as shown in Figure 4 below. The human subjects are also increased from 1 human to 5 humans periodically.



(a) (b)

(c) (d)

(e)

Fig 4. Visualization of LiDAR scanning in the presence of a human subject. The human is situated in Fig 4(a) 1m, Fig 4(b) 2m, Fig 4(b) 3m, Fig 4(c) 4m, and Fig 4(d) 5m respectively from the LiDAR sensor, represented on a single plane.

The features extracted are then trained with machine learning algorithms for detection and recognition purposes. Detailed results of the test and experiments are discussed in chapter four of the paper. The second prototype implied hardware improvement and calibration with the TurtleBot. The detection system is now integrated with the TurtleBot model, implementing an upgraded microcontroller of Raspberry Pi 3. The system for navigation depends on Robot Operating System (ROS) software which enables the navigation and mapping through Simultaneous Localization and Mapping (SLAM). Figure 5 shows the TurtleBot with LiDAR sensors as its detection system.

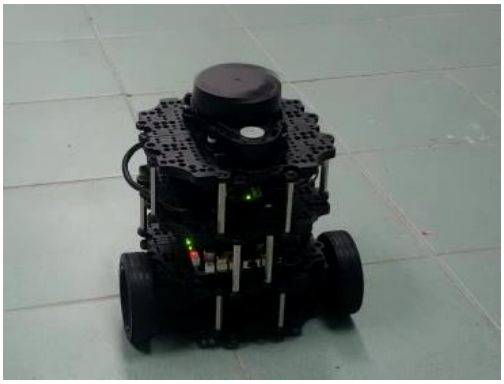


Fig 5. TurtleBot with LiDAR-based sensor

The mobile robot navigation system has been experimented within an approximately 5m × 2m workspace. Mobile robots are required to move towards a conflicting destination without communication and collision with each other. The time taken to complete the route, scanning of the surrounding, a number of collisions, type of obstacles and the number of pathing alterations are recorded for result analysis.

Figure 6 shows the navigation test conducted on the mobile robots. The result shows the robot to arrive towards the destination within 1 minute and 30 seconds, with a single dynamic obstacle, no sudden stop and 6 path alterations recorded. The dynamic obstacle and conflicting destination points are selected as it resembles the uncertaint influence of the real world surrounding [38].

(a) **(b)** **(c)**

Fig 6. Mobile robot navigation shows Fig 6 (a) initial point, Fig 6 (b) meeting path and (c) destination point.

4. Human Recognition

Two experiments were done to verify the performance of the proposed method. The first experiment was comparing the classification trained by different machine learning algorithms with varying parameters. The second experiment is to compute precision, recall and F_1-score and to compare it with other classifiers.

For the extraction through the CE method, we initialize the classifier training on k-NN, Decision Tree method and CNN. The training algorithms are fixed with varying parameters to determine which parameter will yield the best results for the classification task. The experiment was done on an Intel(R) Core (TM) i7-5500U CPU @ 2.40GHz, 8Gb RAM and a 64-bit operating system. The results of the training and testing accuracy of the simulation are represented in Table 2.

Table 2. Result of training algorithm for k-NN, DT and CNN.

Training	Parameter	Value	Accuracy (/%)	Time (s)
k-NN	k	3	78.26	0.013
		6	73.91	0.016
		10	69.56	0.011
DT	Max Depth	1	91	0.004
		3	91	0.007
		5	83	0.007
CNN	No of Layer	53.44	53.44	281
		67.00	67.00	334
		63.67	63.67	365

The parameters in the columns in Table 1 are chosen to identify the best configuration. For k-NN, the varying parameter to be adjusted is the value of k; where k = 3, 6 and 10. Whereas for the Decision Tree algorithm, we set the maximum depth value to be 1, 3 and 5 respectively. And finally, for CNN which is emerging nowadays in numerous classification problems [39], [40], we set the number of hidden layers to be 3, 4 and 5. The batch size is set at 10, with the number of epochs set at 1000, an activation function of Rectified Linear Units (ReLUs) and Softmax function respectively.

Based on the result, the DT algorithm with a maximum depth of 3 recorded the highest testing accuracy with 91%. This is obviously better than k-NN with an accuracy average of 73.91% and CNN with an average accuracy of 61.37%. In terms of the time of execution, the DT algorithm also took the shortest time of completion, followed by k-NN and CNN by a distant margin. Figure 2 shows the tree plot architecture of our DT algorithm.

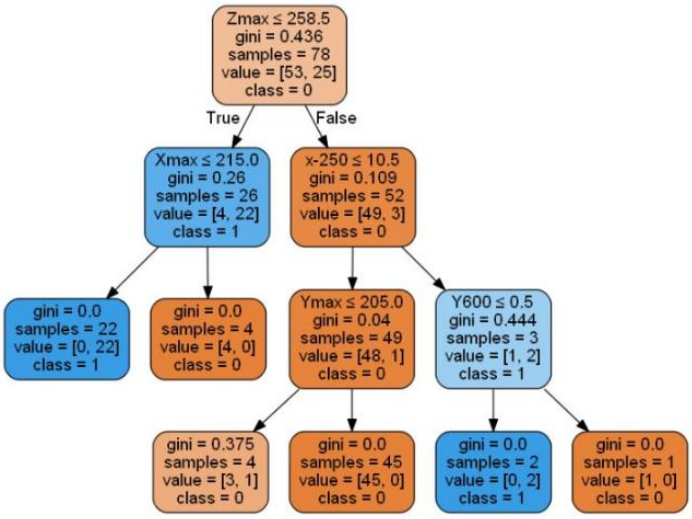


Fig 7. Tree plot of DT architecture.

The Attribute Selection Measure (ASM) or the splitting rule used is the Gini Index, with the "best" split for the splitter parameter. The "gini" score shows the Gini ratio, which measures the impurity of the nodes. A fully grown decision tree without maximum depth tends to overfit the training data. Hence, as the acquired accuracy from Table 1 suggests, we select the maximum depth to be at the value of 3. There are 4 internal nodes and 6 leaf nodes, with values of *zmax*, *xmax* and *ymax* considered as the decision rule for data splitting.

As the CE method trained with a DT algorithm shows the most promising result, we analyse its performance in terms of precision, recall and F_1 score. The number of human and non-human clusters are balanced. Table 2 shows the precision, recall and F-1 score of the DT algorithm based on selected test set data. There is a balance number of random samples taken from human and non-human subjects as can be seen in the "Support" column below.

Table 3. Precision, Recall and F-1 score of human detection

	Precision	Recall	F1-score	Support
Non-Human	88%	88%	87%	12
Human	82%	82%	87%	11
Micro Average	87%	87%	87%	23
Macro Average	87%	87%	87%	23
Weighted Average	87%	87%	87%	23

Based on Table 3, the test set data it can be seen that the detection of human and non-human object clusters are in balance as good scores of precisions, recall and F_1 score are achieved. Precision measures positive predictive values and recall indicate the sensitivity of the predictions. With a score of 82% and 87% for the human and non-human classes respectively, the classification shows decent precision and recall. An excellent F_1 score of 87% shows a harmonic means balance between precision and recall, taking into account both false positives and false negatives. Consistent macro and micro averages were recorded at a score of 87%. Macro-average shows the performance overall across the sets of data and micro-average represents evaluation when the dataset varies in size. This proves that the system represents good performance across all sets of data

5. Conclusion

In this paper, we propose a novel non-end-to-end classifier for sparse LiDAR point cloud human detection. As shown in Table 1, research done on sparse LiDAR data still has room for enhancement and improvement in terms of multi-frame implementation. For our research, we gather authentic point clouds from 100 scenes with human and non-human presences. The data are collected within indoor and outdoor, day and night surroundings, and different backgrounds for better simulating different scenarios of the surrounding. We also analyse the performance of our method with varying algorithms to improve the performance of human detection. The result shows promising state-of-the-art achievement within sparse LiDAR point cloud data.

For future research, distance measurement and position tracking can be implemented on the human detection method. Additional elements or subjects for detection can also be added to further test and improve the reliability of the system.

6. Acknowledgements

The author would like to thank the Ministry of Education Malaysia (MOE) and International Islamic University Malaysia (IIUM) for their support. The research is funded by grant number SRCG20-046-0046 IoT-Based Visually Impaired Community (VIC) Geospatial Tracking with Swarming RoVision (SR), (Collaborative research between IIUM – UiTM – UMP – Southampton University Malaysia).

7. References

[1] Y. Wang, W. L. Chao, Di. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, vol. 2019-June, pp. 8437–8445, doi: 10.1109/CVPR.2019.00864.

[2] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D Object Detection for Autonomous Driving," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7644–7652.

[3] Q. Li *et al.*, "LO-Net : Deep Real-time Lidar Odometry," 2019.

[4] G. Daniel, M. Wang, M. Schn, and T. Ganjineh,

"Radar / Lidar Sensor Fusion for Car-Following on Highways," in *The 5th International Conference on Automation, Robotics and Applications*, 2011, pp. 407–412.

[5] P. P. Neumann and M. Bartholmai, "Real-time Wind Estimation on a Micro Unmanned Aerial Vehicle Using its Inertial Measurement Unit," *Sensors Actuators A. Phys.*, vol. 235, pp. 300–310, 2015, doi: 10.1016/j.sna.2015.09.036.

[6] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple LIDARs and cameras on autonomous vehicles," *Rob. Auton. Syst.*, pp. 1–12, 2016, doi: 10.1016/j.robot.2016.05.010.

[7] K. Banerjee, D. Notz, J. Windelen, S. Gavarraju, and M. He, "Online Camera LiDAR Fusion and Object Detection on Hybrid Data for Autonomous Driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1632–1638.

[8] F. Wotawa and M. Zimmermann, "Adaptive System for Autonomous Driving," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2018, pp. 519–525, doi: 10.1109/QRS-C.2018.00093.

[9] J. Qiu *et al.*, "DeepLiDAR : Deep Surface Normal Guided Depth Prediction for Outdoor Scene from Sparse LiDAR Data and Single Color Image," 2019.

[10] Z. Wang, W. Zhan, and M. Tomizuka, "Fusing Bird ' s Eye View LIDAR Point Cloud and Front View Camera Image for 3D Object Detection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1–6.

[11] Y. Yeniaydin and K. W. Schmidt, "Sensor Fusion of a Camera and 2D LIDAR for Lane Detection," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019, pp. 1–4.

[12] M. R. M. Romlay, M. I. Azhar, and S. F. Toha, "Two-wheel Balancing Robot ; Review on Control Methods and Experiment," vol. 5, no. 6, pp. 106–112, 2017.

[13] H. Ogawa, K. Tobita, K. Sagayama, and M. Tomizuka, "A Guidance Robot for the Visually Impaired : System Description and Velocity Reference Generation," in *Computational Intelligence in Robotic Rehabilitation and Assistive Technologies (CIR2AT)*, 2014, pp. 9–15, doi: https://doi.org/10.1109/CIRAT.2014.7009735.

[14] H. Machiraju and S. S. Channappayya, "An Evaluation Metric for Object Detection Algorithms in Autonomous Navigation Systems and its Application to a Real-Time Alerting System," *2018 25th IEEE Int. Conf. Image Process.*, pp. 81–85, 2018.

[15] K. Ogura, Y. Yamada, S. Kajita, H. Yamaguchi, T. Higashino, and M. Takai, "Ground object recognition and segmentation from aerial image-based 3D point cloud," *Comput. Intell.*, pp. 1–18, 2019, doi: 10.1111/coin.12232.

[16] M. Lehtomäki *et al.*, "Object Classification and Recognition From Mobile Laser Scanning Point Clouds in a Road Environment," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 1226–1239, 2015.

[17] S. Valero, P. Salembier, and J. Chanussot, "Object Recognition in Hyperspectral Images using Binary Partition Tree Representation," *Pattern Recognit. Lett.*, vol. 56, pp. 45–51, 2015, doi: 10.1016/j.patrec.2015.01.003.

[18] Y. Demir, "Object Recognition and Detection with Deep Learning for Autonomous Driving Applications," *Simul. Trans. Soc. Model. Simul. Int.*, vol. 93, no. 9, pp. 759–769, 2017, doi: 10.1177/0037549717709932.

[19] W. Min, Y. Zhang, J. Li, and S. Xu, "Recognition of Pedestrian Activity based on Dropped-Object Detection," *Signal Processing*, vol. 144, pp. 238–252, 2017, doi: 10.1016/j.sigpro.2017.09.024.

[20] L. Zhang, L. L. B, X. Liang, and K. He, "Is Faster R-CNN Doing Well for Pedestrian Detection ?," in *Computer Vision-ECCV 2016*, 2016, pp. 443–457, doi: 10.1007/978-3-319-46475-6.

[21] B. Li, "3D Fully Convolutional Network for Vehicle Detection in Point Cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1513–1518.

[22] A. Kumar, K. Anders, L. Winiwarter, and B. Hofle, "Feature Relevance Analysis for 3D Point Cloud Classification using Deep Learning," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Geospatial Week 2019*, 2019, vol. IV-2/W5, no. June, pp. 10–14.

[23] 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)*, 2017, pp. 652–660.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet ++ : Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 1–10.

[25] D. Maturana and S. Scherer, "VoxNet : A 3D Convolutional Neural Network for Real-Time Object Recognition," in *IEEE/RSJ International COnference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[26] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic Segmentation of 3D Point Clouds," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 537–547.

[27] A. Börcs, B. Nagy, and C. Benedek, "Instant Object Detection in Lidar Point Clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 992–996,

2017.

[28] A. Mandow, G. J. Antonio, and A. J. Garc, "Optimizing Scan Homogeneity for Building Full-3D Lidars based on Rotating a Multi-Beam Velodyne Range-finder," in *IEEE/RSJ International Conference on Intelligent robots and Systems (IROS)*, 2018, pp. 4788–4793.

[29] Y. Ye, H. Chen, C. Zhang, X. Hao, and Z. Zhang, "SARPNET: Shape Attention Regional Proposal Network for LiDAR-Based 3D Object Detection," *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2019.09.086.

[30] A. Husain and R. Chandra Vaishya, "Road Surface and its Center Line and Boundary Lines Detection using Terrestrial Lidar data," *Egypt. J. Remote Sens. Sp. Sci.*, vol. 21, no. 3, pp. 363–374, 2018, doi: 10.1016/j.ejrs.2017.12.005.

[31] J. Wu, Y. Tian, H. Xu, R. Yue, A. Wang, and X. Song, "Automatic Ground Points Filtering of Roadside LiDAR Data using a Channel-based Filtering Algorithm," *Opt. Laser Technol.*, vol. 115, pp. 374–383, 2019, doi: 10.1016/j.optlastec.2019.02.039.

[32] B. Lv, H. Xu, J. Wu, Y. Tian, S. Tian, and S. Feng, "Revolution and Rotation-based Method for Roadside LiDAR Data Integration," *Opt. Laser Technol.*, vol. 119, p. 105571, 2019, doi: 10.1016/j.optlastec.2019.105571.

[33] M. P. Muresan, S. Nedevschi, and I. Giosan, "Real-Time Object Detection Using a Sparse 4-Layer LIDAR," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2017, pp. 317–322.

[34] Y. Tatebe, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, and U. Sakai, "Pedestrian Detection from Sparse Point-Cloud using 3DCNN," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018, pp. 1–4.

[35] B. Xie, Y. Gu, Q. Wang, and H. Liu, "LiDAR Point Classification Based on Joint Sparse Representation in Kernel Space," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 1476–1479.

[36] Raja Kushaslnagar, S. Ludi, and P. Kushalnagar, "Multi-View Platform : An Accessible Live Classroom Viewing Approach for Low Vision Students," in *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, 2011, pp. 267–268.

[37] M. M. Rashid, M. R. M. Romlay, and M. M. Ferdaus, "Development of Electronic Rain Gauge System," *Int. J. Electron. Electr. Eng.*, vol. 3, no. 4, pp. 245–249, 2015, doi: 10.12720/ijeee.3.4.245-249.

[38] A. M. Ibrahim, M. Saifullah, M. R. M. Romlay, I. Venkat, and I. Ibrahim, "Hybrid Social Force-Fuzzy Logic Evacuation Simulation Model for Multiple Exits," in *2019 7th International Conference on Mechatronics Engineering, ICOM 2019*, 2019, pp. 1–5, doi: 10.1109/ICOM47790.2019.8952063.

[39] M. R. M. Romlay, M. M. Rashid, and S. F. Toha, "Development of Particle Swarm Optimization Based Rainfall-Runoff Prediction Model for Pahang River , Pekan," *2016 Int. Conf. Comput. Commun. Eng.*, pp. 306–310, 2016, doi: 10.1109/ICCCE.2016.72.

[40] M. Rabani, M. Romlay, M. M. Rashid, and S. F. Toha, "Rainfall-Runoff Model Based on ANN with LM , BR and PSO as Learning Algorithms," *Interational J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 971–979, 2019, doi: 10.35940/ijrte.C4115.098319.