Unsupervised Clustering-based 3D Static Scene Construction Using LiDAR Channel and Azimuth Angle

Rohit Rajput, Salil Goel, Aditya Medury

Department of Civil Engineering, IIT Kanpur, Uttar Pradesh, India - rrajput@iitk.ac.in, sgoel@iitk.ac.in, amedury@iitk.ac.in

Keywords: Light Detection and Ranging (LiDAR), Static Roadside LiDAR, 3D Static Environment, DBSCAN, Azmuth, Laser Channel

Abstract

Cameras are typically used at road intersections to collect data to perform object detection but struggle in low-light and harsh weather. On the other hand, Light Detection and Ranging (LiDAR) is used as a key technology in 3D vision systems. It gives the 3D point cloud, which includes accurate depth information, but its resolution is cost-dependent, with higher resolutions being more expensive. Deep learning-based method requires large, labelled dataset which increases the cost, time and accuracy depending on the model trained on labelled dataset. To perform object detection in point cloud data is a challenging task due to the incomplete representations, data sparsity and unavailability of training data. To overcome this by using an unsupervised approach, it is important to identify the static scene and then detect the moving object. This work incorporated a novel approach to construct static scene using azimuth angle and laser channel information using an unsupervised clustering approach. This work incorporates two modules I) data collection using VLP-16 LiDAR, and II) static scene construction using DBSCAN (density-based spatial clustering and noise) clustering-based approach. Data is collected at a 4-legged intersection and pre-processed to extract aggregated distances corresponding to the unique pair of azimuth angle and laser channel. DBSCAN is used to perform clustering on the aggregated distances, based on the highest silhouette score and lowest intra distance between points in cluster, static points are identified, and static scene constructed. The qualitative evaluation of method demonstrates that the algorithm effectively and accurately filters out background points.

1. Introduction

In recent decades, intersection safety and efficiency have become focal points for researchers, driving the implementation of cutting-edge technologies. Road intersections are essential parts of transportation infrastructure since they are the locations at which automobiles, pedestrian and cyclists interact, making them potential hotspots for accidents and congestion. At intersections, objects differ in size, including smaller objects such as road boundaries, curbs, and statues, as well as larger objects such as trees, poles, and buildings. Accurate and timely detection of these objects are critical for ensuring safe and efficient traffic flow. These objects are categorized as static and part of the static scene. Constructing static scene can provide traffic engineers and city planners valuable information to optimize traffic flow and improve safety. In static scenes, objects show minimal or no motion over the observed timeframe, and their spatial characteristics remain constant. Various sensors are used to capture static object data such as cameras, RADAR (radio detection and ranging), and LiDAR (light detection and ranging). Among these, LiDAR stands out as a pivotal tool within vision systems, enabling the detection of road objects in three-dimensional space. LiDAR sensors are widely used in autonomous driving applications for object detection and mapping, with the scope of the application currently limited to a road intersection. Although cameras are typically used at road intersections to collect data to perform object detection, provide high-resolution images but struggle in low-light and harsh weather (Al-Haija et al.,2022). On the other hand, LiDAR gives the 3D point cloud, which includes accurate depth information, but its resolution is cost-dependent, with higher resolutions being more expensive (Benedek et al.,2021). LiDAR boasts the ability to provide precise 3D point cloud data under varying weather conditions and lighting scenarios, offering comprehensive 360-degree coverage of intersections. Within this captured point cloud data, both static and dynamic objects are present. To enhance intersection safety and efficiency, roadside systems can potentially use this data to detect road objects and track their movements, extracting valuable trajectory information. However, the density of the point cloud presents a challenge, as the moving object point density changes when the object is far from sensor. This variability affects the performance of moving object detection algorithms, increasing computational costs, processing times, and affecting overall accuracy (Liu et al.,2022). To address this issue, a common approach involves identifying static object points and constructing a static scene (Zhao et al., 2019). Objects in point cloud data can be identified using machine learning (ML) techniques, such as supervised, unsupervised, and deep learning (DL) (Charles et al., 2018). Supervised and DL methods need training data (such as labelled objects) to train algorithms to perform object detection (Zhang et al., 2015) (Sakkos et al., 2017) (Babaee et al., 2018). The challenges associated with the training data is limited availability, sensor, and location specific. On the other hand, instead of using training data unsupervised methods used cluster-based approach (K-means clustering (Tonini et al.,2014), DBSCAN, spectral clustering etc. (Murugesan et al.,2021)) to identify objects. DBSCAN is advantageous over K-means clustering as it does not require predefined number of clusters and instead depends on two parameters which is minimum number of points and the radius. The point clouds are sparse and unstructured and very few points represent the object of interest in a frame.

Static scene construction plays a vital role in identifying object of interest from the point clouds. There are various methods used to identify static points like azimuth-height table (Zhao et al., 2019), vertical angle and horizontal azimuth angle of the LiDAR beam (Lee et al., 2012), background difference method (Zheng et al., 2021), variable-dimension background filtering method (Wu et al., 2021), elevation azimuth matrix (Zhang et al., 2022), adaptive grid (Wang et al., 2022). Static scene

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-381-2024 | © Author(s) 2024. CC BY 4.0 License.

construction typically involves three main approaches: reference-based methods (Zhang et al., 2022), voxelization-based methods (Asvadi et al.,2016), and point-based methods, with the latter subdivided into raw point cloud (Zhao et al., 2019) and image-based methods (Chen et al.,2023). Reference-based methods (Zheng et al., 2021) involve manual choice of frames devoid of moving objects, where all points are considered as background points. However, this approach struggles to accurately find background points due to point fluctuation. It is overcome by aggregating multiple frames, but it is not possible to find multiple frames without any moving object (Tarko et al.,2018). Reference-based methods are the simplest methods. The voxelization-based method (Asvadi et al.,2016), also known as rasterization methods, converts 3D space into small cubes within aggregated frames and constructs static scenes based on voxel density. 3D-DSF (Wu at al.,2017), voxel-based background filtering method uses aggregated LiDAR frames which converts the 3D space into the small cubes and based on the point distribution, static points are identified using predefined threshold. It experiences performance degradation in congested scenarios or when moving objects come to a halt, with accuracy depends on the size of the cubes. Voxel-based methods work well in free flow traffic conditions also it solves the problem related to manual selection of frames by choosing random frames, but challenge is the number of voxels directly proportional to the computational load. Selection of voxel size also impacts the accuracy of the methods. Raw point-based method identifies background point directly from the raw data without any conversion which reduces the computational load, cost, and time. It uses the nearest neighbours (NN), azimuth-height table (Zhao et al., 2019), and laser channels to map the static point across the frames but performance reduces in adverse weather conditions. Range Image-based method (Chen et al., 2023) converts the raw point cloud in the form of range image and analyses the change pixelwise to create background scene. Depth information is missing from the image-based methods (Wu et al.,2022).

Based on the literature, several research gaps have been identified that impact the field of static scene construction from sensor data. Firstly, there is an ambiguity in frame selection, which hinders the transferability of methods due to parameter selection challenges, affecting the accuracy of static scene construction. Additionally, the absence of methods for constructing static scenes using multi-sensor data limits the ability to conduct comprehensive analyses. This is further compounded by the lack of well-defined ground truth, which impedes the quantification of results and makes it difficult to assess the effectiveness of different approaches.

This paper aims to develop and evaluate a novel method for constructing 3D static scenes that requires minimal parameter fine-tuning, eliminating the necessity for manual parameter adjustment for the LiDAR point cloud data. This method directly processes raw 3D point cloud by selecting points from a randomly chosen set of frames within the LiDAR data. We extract the point distance for each azimuth angle and laser channels. This information is used to create a 2D matrix with N rows, representing the number of laser channels (C) and M columns represents azimuth values (α), effectively spanning 360° at an interval of 0.2°. The distances are aggregated across frame and clustered using DBSCAN to identify static scene. The paper presents a novel approach based on azimuth angle and laser channel for 3D static scene creation without the need for manual parameter finetuning and the results will undergo precise qualitative evaluation and comprehensive comparison to ensure the method's accuracy and reliability.

The remainder of this paper is structured into following sections: Section II gives the overview of the related works. Section III presents the methodology which includes the data collection, data preprocessing and static scene construction using DBSCAN clustering. Section IV evaluates and compares the accuracy of the proposed methodology. Section V concludes the paper and discusses the future scope of this research work.

2. Methodology

To construct a static scene from the LiDAR point cloud data, the methodology adopted is illustrated in Figure 1. It processes LiDAR data, beginning with data collection using LiDAR sensors to measure distances by reflecting laser light off objects. The process includes data preprocessing to clean and standardize the incoming data, followed by frame segmentation to analyse each frame individually. An elementwise 2D distance matrix is then formed for each frame, capturing distance measurements based on azimuth angle and laser ID. Subsequent steps involve aggregating distance data across frames and

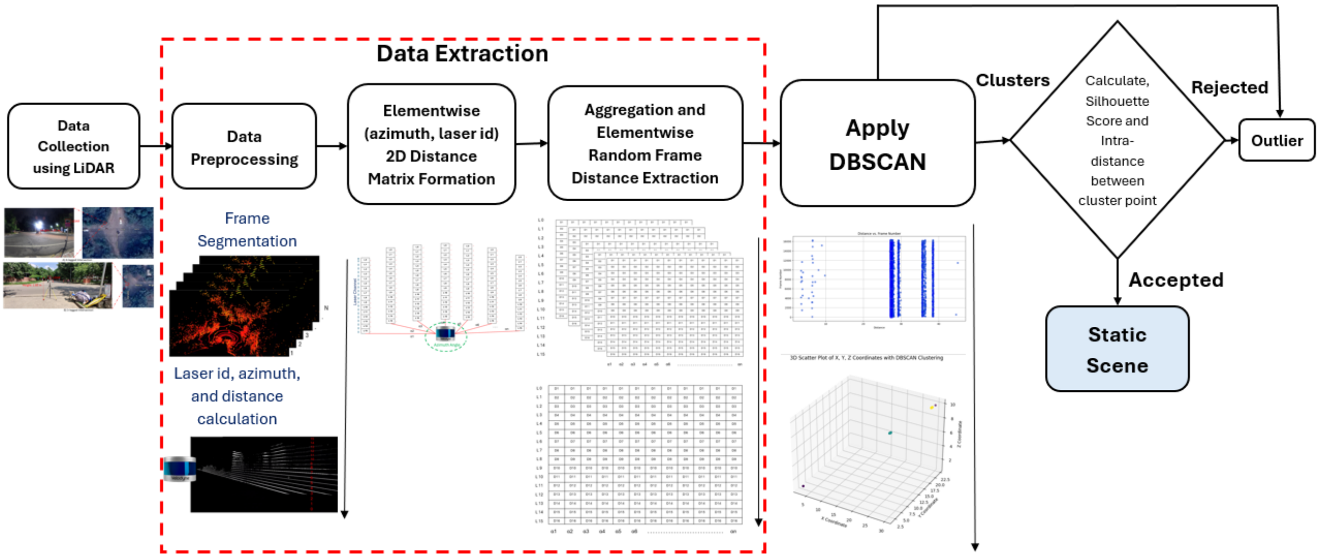

Data Extraction
Data Collection using LiDAR
Data Preprocessing
Elementwise (azimuth, laser id) 2D Distance Matrix Formation
Aggregation and Elementwise Random Frame Distance Extraction
Apply DBSCAN
Clusters
Calculate, Silhouette Score and Intra-distance between cluster point
Rejected
Outlier
Accepted
Static Scene
Frame Segmentation
Laser id, azimuth, and distance calculation

Figure 1. Flow Chart of LiDAR-Based Clustering and Background Point Filtering Using DBSCAN

applying the DBSCAN clustering algorithm to identify dense clusters and outliers within the data. Silhouette score and distance between points in cluster is calculated to identify static points. In the next section detailed description of each step is given.

2.1 Data Collection

LiDAR is the only sensor used for data collection in this work. There are multiple types of LiDAR sensor available in the market which are used for vision tasks. In this work velodyne VLP-16 model is used for data collection. VLP-16 is a 16-channel small, cylindrical sensor installed at the corner of a road intersection. It offers 360-degree horizontal field-of-view (HFOV) and 30-degree (-15 to +15 degree) vertical field-of-view (VFOV) with a range of up to 100 meters shown in Figure 2A. It captures up to 3,00,000 points per second and has horizontal resolution 0.1-0.4-degree, vertical resolution 2-degree and operating frequency varies between 5-20Hz. It collects data in challenging environmental conditions and is suitable for outdoor conditions for extended periods of data collection. VLP-16 uses the time-of-flight (ToF) method to measure distance. As shown in Figure 2b, LiDAR emitter emits the laser at a horizontal angle α (top view) and vertical angle ω (front view), receiver receives the transmitted signal and calculates the distance R by equation 1 where c is the speed of the light already known and Δt is the time taken by the laser pulse between the emitted to reflected by the target.

$$R = \frac{c\Delta t}{2} \tag{1}$$

By using the formula given in figure 2B, point coordinate value in X, Y, Z direction is calculated. VLP-16 has its own coordinate frame in which origin lies 3.77 cm above the base and X, Y are the horizontal axis and Z is the vertical axis (Figure 2b).

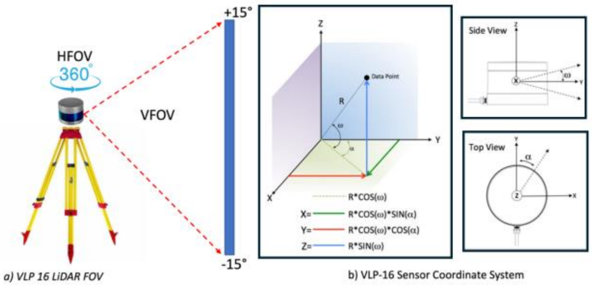


Figure 2. a) LiDAR Properties (FOV) and b) Coordinate Frame

Figure 3 shows the sensor suite used for the data. The hardware part consists of 1 velodyne vlp-16 LiDAR sensor, 1 tripod on which sensor is mounted, LiDAR interface box, 1 power bank 60000 mAh, 1 DC-to-AC converter for constant power supply to VLP-16 and 1 laptop. The software part uses a Linux-based operating system (Ubuntu 20.04) compatible with ROS (robot operating system) noetic which provide interface to control LiDAR sensor, visualize data in real time and store the collected data. Data are stored in the form of rostopic pointcloud2 messages which contains X, Y, Z, I, and timestamp. As the vertical angle is fixed to 2°, distance and azimuth angle is calculated using the formula given in Figure 2b.

Two types of intersections are selected shown in Figure 4, one is a 4-legged intersection (Figure 4a) which is used for analysis in this paper and other is a 3-legged intersection (Figure 4b). Data is collected during peak hours, daytime, and nighttime. Duration of data collection is more than 5 hours. Data is collected in static mode wherein sensor is fixed on the tripod and located at the corner of the intersection.

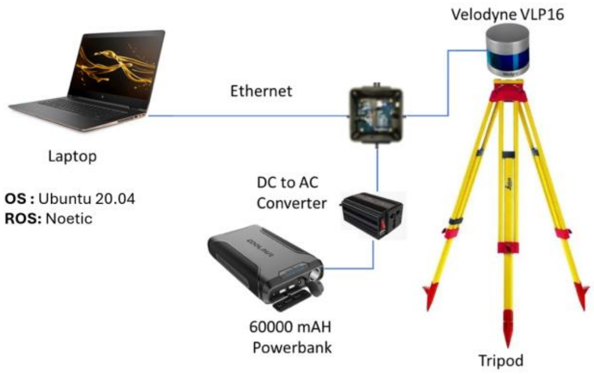


Figure 3. Sensor setup

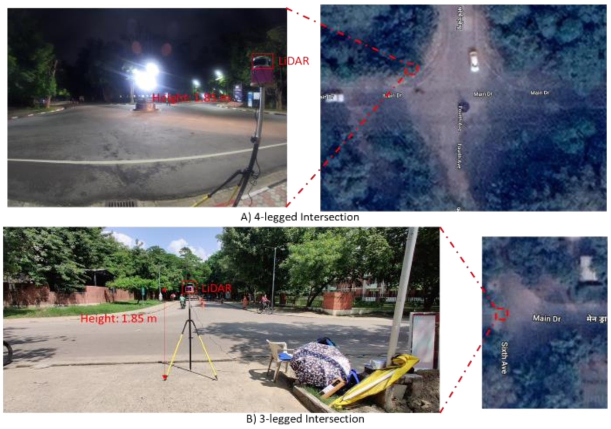


Figure 4. Location a) 4-legged Intersection b) 3-legged Intersection

2.2 Data Extraction

Data collected from the velodyne VLP-16 is saved using ROS in the .bag file format. In .bag format each scan from the vlp-16 recorded in the form topic point cloud2. Raw file contains the point cloud information (X, Y, Z, and I), horizontal angle (α), vertical angle (ω) and timestamp of the scan. Each value of point occupies 16 bytes memory. It is difficult to process all points at a once, so we extract each frame from the raw data.

At each azimuth angle, 16 lasers fired by the sensor and the corresponding values are recorded by the sensor and stored. The vertical angle of the laser id is fixed, and azimuth resolution depends on the rotation speed of sensor. Figure 5 shows the raw frame which is generated after sensor completes one rotation. In total 16,178 frames are used to construct a static scene.

Figure 6 shows the extraction of distances (m) for each pair of laser id and azimuth angle for each frame. The number of elements calculated based on the laser channel (C), sensor rotation speed (or azimuth resolution, α_{res}) and calculated using the following formula.

$$Number\ of\ Elements = C \times \frac{360°}{\alpha_{res}} \tag{2}$$

https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-381-2024 | © Author(s) 2024. CC BY 4.0 License.

Based on the formula, VLP 16 fires 16 lasers in 0.2° azimuth resolution when its rotation frequency is 600 rpm, resulting in 1800 elements.

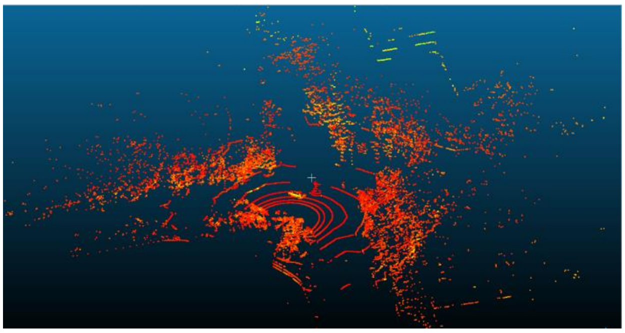


Figure 5. Raw Point Cloud Frame

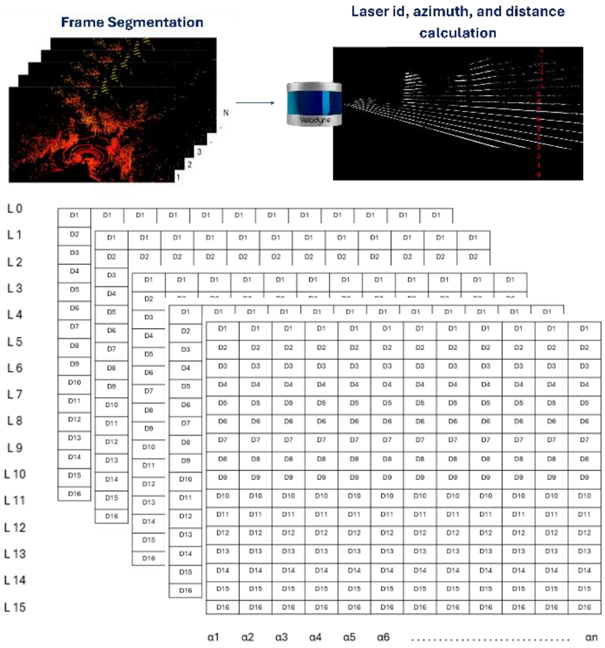
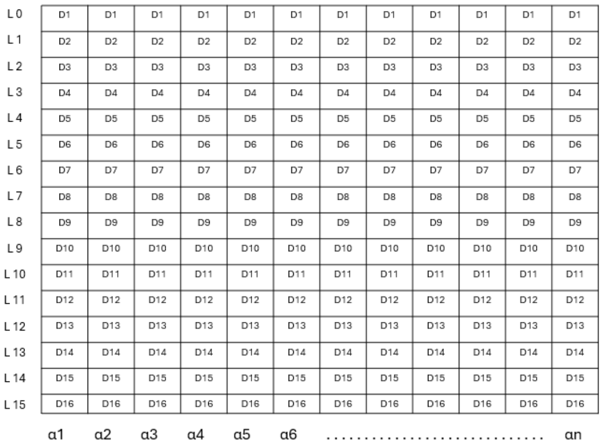


Figure 6. Framewise distance extraction



	α1	α2	α3	α4	α5	α6	....					αn
L 0	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1
L 1	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2
L 2	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3
L 3	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4
L 4	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5
L 5	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
L 6	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7
L 7	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8
L 8	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9
L 9	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10
L 10	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11
L 11	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12
L 12	D13	D13	D13	D13	D13	D13	D13	D13	D13	D13	D13	D13
L 13	D14	D14	D14	D14	D14	D14	D14	D14	D14	D14	D14	D14
L 14	D15	D15	D15	D15	D15	D15	D15	D15	D15	D15	D15	D15
L 15	D16	D16	D16	D16	D16	D16	D16	D16	D16	D16	D16	D16

Figure 7. Elementwise aggregated distance matrix

A frame-wise distance matrix is developed as shown in figure 6. This will further process to get static cluster value for each element of distance matrix. Further frames are aggregated to get aggregated distance matrix shown in Figure 7.

2.3 Static Scene Construction

The main purpose of static scene construction is to retain the points of object of interest in the point cloud data as much as possible. As shown in Figure 8, the process to construct static scene using DBSCAN clustering begins with selecting and analysing each element of aggregated distance matrix for the given channel data (C, α) and iterate through all the elements individually.

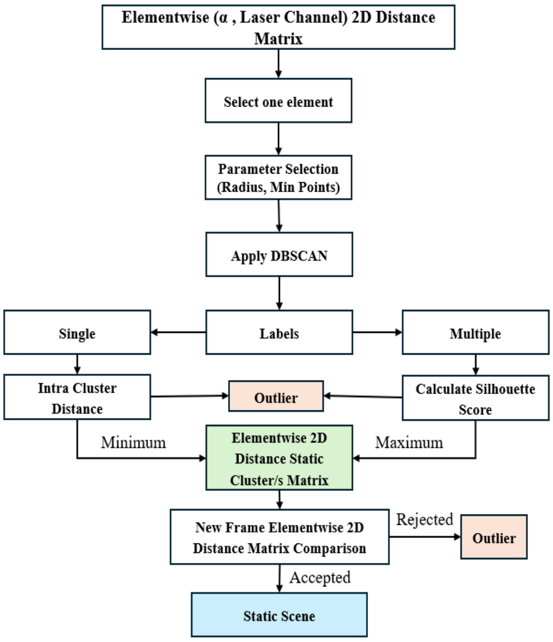

Elementwise (α , Laser Channel) 2D Distance Matrix
Select one element
Parameter Selection (Radius, Min Points)
Apply DBSCAN
Single
Labels
Multiple
Intra Cluster Distance
Outlier
Calculate Silhouette Score
Minimum
Elementwise 2D Distance Static Cluster/s Matrix
Maximum
New Frame Elementwise 2D Distance Matrix Comparison
Rejected
Outlier
Accepted
Static Scene

Figure 8. Flow Chart to Identify Static Scene

The next step involves determining the appropriate parameters for the DBSCAN algorithm, specifically the radius (ε) and the minimum number of points (MinPts) required for forming a cluster. Once the parameters are set, the DBSCAN clustering algorithm is applied to the selected element. The results from DBSCAN are then used to assign labels to the data points, categorizing them into either a single cluster or multiple clusters. If a single cluster is detected, the intra-cluster distance is calculated to understand the compactness of the cluster. If multiple clusters are detected, the silhouette score for each cluster is computed to assess the clustering quality and the separation between clusters. Outliers within the clusters are identified as data points that do not fit well into any cluster. An elementwise 2D distance static clusters matrix is then constructed, incorporating both the identified clusters and outliers. The identified clusters are combined to construct static scenes from the aggregated frames. This structured approach ensures a thorough analysis of the 2D distance matrix, enabling effective clustering and outlier detection using the DBSCAN algorithm. After selecting an element from the aggregated distance matrix, next step is to select the appropriate parameters which is used in DBSCAN clustering to get clusters. The next section explains the selection of parameters.

2.3.1 Parameter Selection and Clustering

This section explains the selection of parameters. Initially, aggregated distance element corresponding to unique pair of azimuth angle and LiDAR channel is selected. Parameters such as total number of points and minimum sample points are calculated subject to 1% of total points or minimum of 100 points to form a cluster, and initial values for eps (eps_initial= 0.08 and eps_max= 0.4) are set. Since the divergence of laser beam increases with the increase in distance because of azimuth resolution and fired laser can hit anywhere on the arc length (S) as shown in the Figure 9 . The permissible deviation for laser hitting at the prescribed arc length is 3.49 mm/m.

$$Arc\ Length\ (S) = D\theta \tag{3}$$

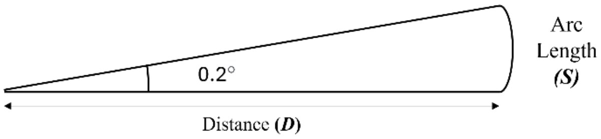


Figure 9. Laser Beam Divergence

The best radius (eps) is found by iterate through the eps, applying DBSCAN, fitting the model, and getting labels based on the best combination of eps and MinPts (Figure 10). If multiple clusters are identified (Figure 11), the intra-cluster distance and silhouette score are calculated, and the best eps is updated based on these metrics. Multiple clusters are observed when objects like trees are present in the scene, as their movement causes the laser to intermittently hit the trees or objects beyond them. Figure 12 shows the spread of aggregated distances for the multiple clusters case. If only one cluster is detected (Figure 13) and all iterations have multiple clusters is set to False, then the intra-cluster distance is calculated. Figure 14 shows the spread of aggregated distances for the single cluster case. If all iterations have multiple clusters, the loop through eps_values are repeated, and the best eps is updated based on silhouette scores. Minimum eps are selected 0.08 and maximum is 0.4. Methods evaluate the cluster at each eps value with the interval of 0.01 Once the best parameters are determined, DBSCAN is applied with this value and min_samples, the model is fitted, and labels are obtained. To find the best value of parameters, the number of clusters is evaluated, and if multiple clusters are present, the silhouette score and intra-cluster distance are calculated; if only one cluster is present, only the intra-cluster distance is calculated.

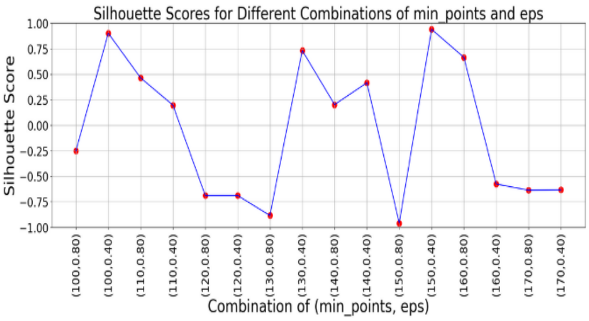


Figure 10 DBSCAN Parameter Selection

To select the best parameter, maximum silhoutte score is selected in case of multiple cluster and minimum intra-cluster point distance is selected in case of single cluster.

After identifying the suitable parameters, DBSCAN clustering is performed to get clusters of the aggregated distances. Figure 15 shows the multiple clusters formed for the laser id 10 and azimuth angle 36 and Figure 16 shows the single cluster formed in laser id 0 and azimuth angle 36.

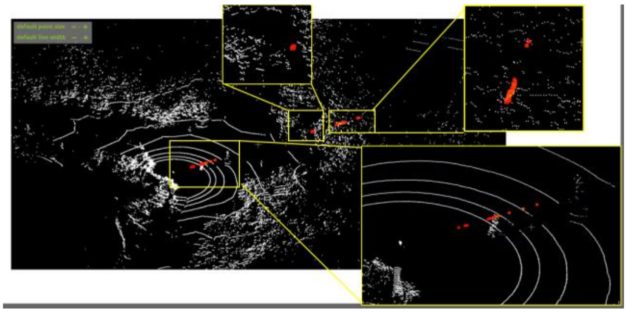


Figure 11. Multiple Clusters Laser Hitting Object (10,36)

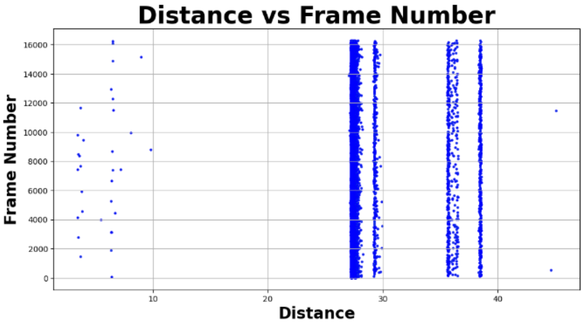


Figure 12. Multi Cluster Distribution of Aggregated Distance Element (10,36)

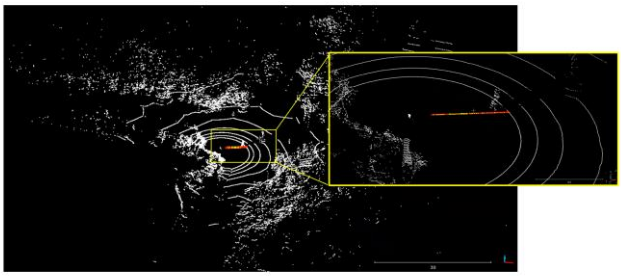


Figure 13. Single Cluster Representation Laser Hitting Object (10,36)

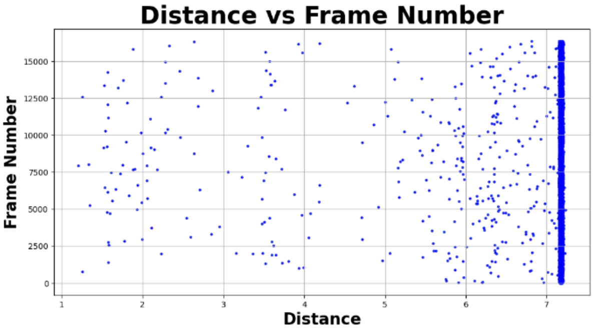


Figure 14. Distribution of Aggregated Distance Element (0,36)

3. Results and Discussion

DBSCAN is performed on each element of aggregated distance matrix and combining them into a frame represents a static scene. Outliers are removed from the combined matrix and remaining points represents static points. Figure 17 is the static scene generated from the method and Figure 18 shows the outlier points identified in single frame. Static points are represented in white colour and outlier in red.

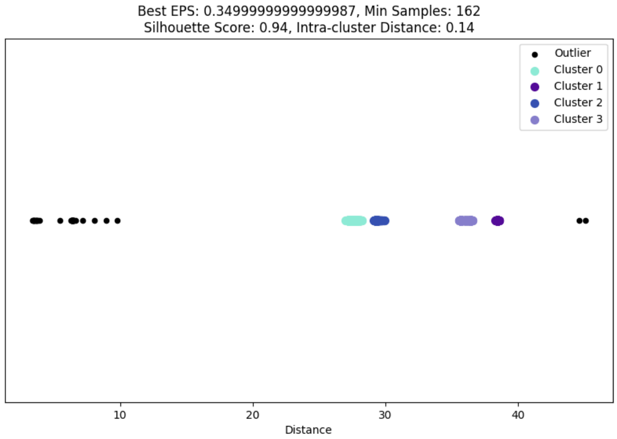


Figure 15. DBSCAN Clustering output (10,36)

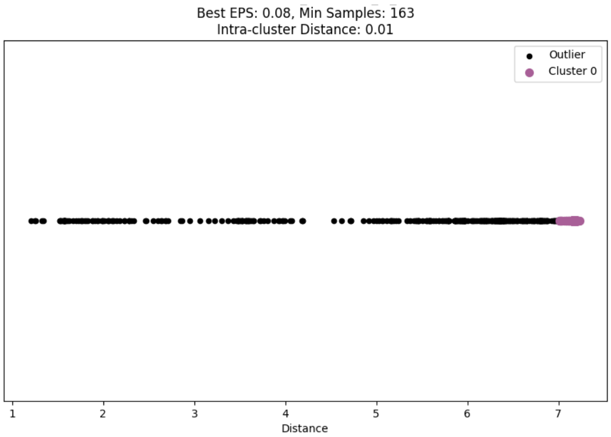


Figure 16. DBSCAN Clustering output (0,36)

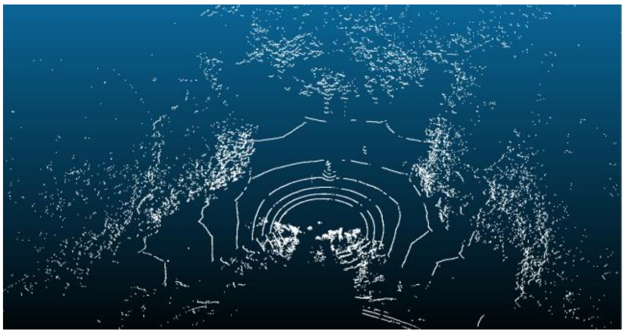


Figure 17. Static Scene Single Frame

By creating static scene framewise from the aggregated frames, there is a problem of occlusion in single LiDAR case as shown in the Figure 19. Qualitative evaluation is done to check the method performance.

Aggregated static points are then identified to construct static scene and Figure 20 shows the static scene generated which shows that the occluded points appeared in the scene and few more points are there (shown in Figure 20 in red box) because of duration of data collection is less. Data collected for longer duration in different traffic scenarios can solve the problem of few outlier points appearing in the static scene.

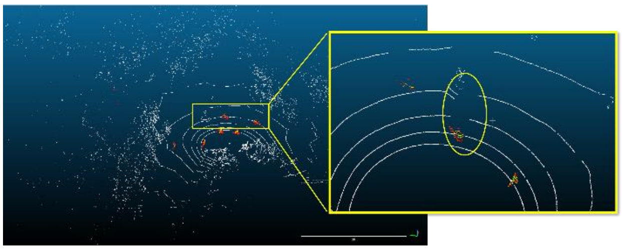


Figure 19. Occlusion in Single Frame

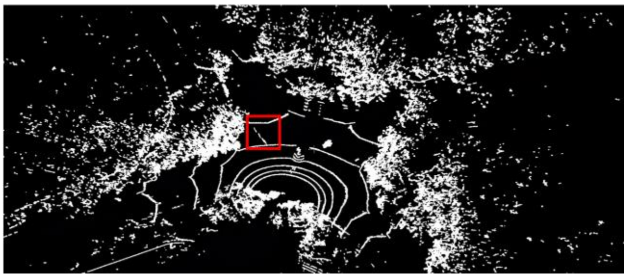


Figure 20. Combined Frames Static Scene

4. Conclusion and future works

In this paper, we developed a novel algorithm to construct static scene using roadside LiDAR based on unsupervised clustering. Static scene construction using unsupervised methods advances the accuracy enhancement of object detection and tracking by reducing unnecessary points. Algorithm removes the ambiguity in frame selection and manual selection of parameters like voxel-based methods by automatically selecting the best parameter for each element (C, α), which increases the transferability of the method. The results show the method effectively and accurately filters out background points. In this work, a single LiDAR is used which offers simplicity, straightforward setup, and cost-effectiveness. But it faces challenges such as limited coverage, and blind spots. Future work includes overcoming the challenges of single LiDAR sensor, multi-lidar setups for enhanced coverage, resolution, and redundancy. Due to the lack of ground truth data, this paper does not include a quantitative evaluation. Future work will address this by creating a static scene using high-resolution LiDAR for such analysis. Future works also include fully automating the process to construct static scene for multiple LiDAR and different type of intersection by considering different traffic scenarios in mixed traffic conditions.

5. Acknowledgements

This work is partially supported by TIH-IoT Chanakya fellowship, and partial support from SERB (Science and Engineering Research Board) and IIT Kanpur. The authors would like to acknowledge the support received from these sources.

References

Al-Haija, Q. A., Gharaibeh, M., & Odeh, A. (2022). Detection in adverse weather conditions for autonomous vehicles via deep learning. *AI*, *3*(2), 303-317. https://doi.org/10.3390/ai3020019

Asvadi, A., Premebida, C., Peixoto, P., & Nunes, U. (2016). 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems, 83*, 299-311. https://doi.org/10.1016/j.robot.2016.06.007

Babaee, M., Dinh, D. T., & Rigoll, G. (2018). A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition, 76*, 635-649. https://doi.org/10.1016/j.patcog.2017.09.040

Benedek, C., Majdik, A., Nagy, B., Rozsa, Z., & Sziranyi, T. (2021). Positioning and perception in LIDAR point clouds. *Digital Signal Processing, 119*, 103193. https://doi.org/10.1016/j.dsp.2021.103193

Chen, Z., Xu, H., Zhao, J., & Liu, H. (2023). A Novel Background Filtering Method with Automatic Parameter Adjustment for Real-Time Roadside LiDAR Sensing System. *IEEE Transactions on Instrumentation and Measurement*. https://doi.org/10.1109/TIM.2023.3300457

Lee, H., & Coifman, B. (2012). Side-fire lidar-based vehicle classification. *Transportation Research Record, 2308*, 173–183. https://doi.org/10.3141/2308-19

Liu, H., Lin, C., Gong, B., & Wu, D. (2022). Extending the detection range for low-channel roadside LiDAR by static background construction. *IEEE Transactions on Geoscience and Remote Sensing, 60,* 1-12. https://doi.org/10.1109/TGRS.2022.3155634

Murugesan, N., Cho, I., & Tortora, C. (2021). Benchmarking in cluster analysis: A study on spectral clustering, DBSCAN, and K-Means. In *Data Analysis and Rationality in a Complex World* (Vol. 16, pp. 175-185). Springer International Publishing. https://doi.org/10.1007/978-3-030-60104-1_20

Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). Frustum pointnets for 3D object detection from RGB-D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 918-927). https://doi.org/10.48550/arXiv.1711.08488

Sakkos, D., Liu, H., Han, J., et al. (2018). End-to-end video background subtraction with 3D convolutional neural networks. *Multimedia Tools and Applications, 77*, 23023–23041. https://doi.org/10.1007/s11042-017-5460-9

Tarko, A., Romero, M., Ariyur, K., Bandaru, V., & Lizarazo, C. (2018). Detecting and tracking vehicles, pedestrians, and bicyclists at intersections with a stationary LiDAR. In *18th International Conference Road Safety on Five Continents (RS5C 2018)*, Jeju Island, South Korea, May 16-18, 2018.

Tonini, M., & Abellan, A. (2014). Rockfall detection from terrestrial LiDAR point clouds: A clustering approach using R. *Journal of Spatial Information Science, 8*, 95-110.

Wang, L., & Lan, J. (2022). Adaptive Polar-Grid Gaussian-Mixture Model for Foreground Segmentation Using Roadside LiDAR. *Remote Sensing, 14*(2522). https://doi.org/10.3390/rs14112522

Wu, J., Xu, H., & Zheng, J. (2017). Automatic background filtering and lane identification with roadside LiDAR data. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-6). IEEE. https://doi.org/10.1109/ITSC.2017.8317723

Wu, J., Lv, C., Pi, R., Ma, Z., Zhang, H., Sun, R., & Wang, K. (2021). A Variable Dimension-Based Method for Roadside LiDAR Background Filtering. *IEEE Sensors Journal, 22*, 832-841. https://doi.org/10.1109/JSEN.2021.3125623

Zhang, Y., Wang, J., Wang, X., Li, C., & Wang, L. (2015). 3D LIDAR-Based Intersection Recognition and Road Boundary Detection Method for Unmanned Ground Vehicle. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems* (pp. 499-504). Gran Canaria, Spain. https://doi.org/10.1109/ITSC.2015.88

Zhao, J., Xu, H., Xia, X., & Liu, H. (2019). Azimuth-Height background filtering method for roadside LiDAR data. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 2421-2426). IEEE. https://doi.org/10.1109/ITSC.2019.8917369

Zhang, T., & Jin, P. J. (2022). Roadside lidar vehicle detection and tracking using range and intensity background subtraction. *Journal of Advanced Transportation, 2022*. https://doi.org/10.1155/2022/2771085

Zheng, J., Yang, S., Wang, X., Xiao, Y., & Li, T. (2021). Background Noise Filtering and Clustering With 3D LiDAR Deployed in Roadside of Urban Environments. *IEEE Sensors Journal, 21*, 20629-20639.