Int. J. Advance Soft Compu. Appl, Vol. 18, No. 2, July 2026
Print ISSN: 2710-1274, Online ISSN: 2074-8523

A Clustering and Convex hull-based approach for Vehicle Bounding Box Localization in LiDAR Images for Advanced Driving Assistance System

Shilpa Ankalaki[1], Geetabai S Hukkeri[1], and Shaleen Bhatnagar[1*]

[1] *Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India*
e-mail:shilpa.ankalaki@manipal.edu
geetabai.hukkeri@manipal.edu
shaleen.bhatnagar@manipal.edu

*Corresponding Author: Shaleen Bhatnagar (shaleen.bhatnagar@manipal.edu)

Abstract

The accurate and efficient localization of vehicle bounding boxes in LiDAR images is of great importance in Advanced Driver Assistance Systems (ADAS) for improving road safety. In this paper, a novel and interpretable method for vehicle bounding box localization in LiDAR images using a combination of point cloud clustering and an efficient convex hull algorithm is proposed. The method starts with the application of a clustering algorithm to LiDAR point cloud images to obtain potential vehicle bounding box information. This involves grouping LiDAR points of individual vehicles, even in the presence of noise and partial occlusion. Next, an efficient convex hull algorithm is applied to obtain a minimal bounding polygon for each cluster of points, representing the geometric approximation of the vehicle. For better bounding box localization accuracy, additional refinement techniques are applied to the method. These techniques include statistical outlier removal, adaptive clustering parameters for varying vehicle sizes, and corrections for occluded objects. The effectiveness of the method was tested using the KITTI dataset, with the method achieving a mean absolute error of 0.167, root mean squared error of 0.186, and an average percentage error of around 8.5% for vehicle dimensions. The approach was also able to attain a Pearson correlation coefficient of 0.9995 with the ground truth data. Additionally, the 1D IoU scores were above 93% on average, which further confirms the excellent spatial overlap. The convex hull-based approach provides several benefits with its simplicity, robustness, and efficiency. It is also model-free and geometry-driven, which is quite beneficial for safety-critical ADAS applications.

Keywords: *Autonomous Vehicle, ADAS, Bounding Box, LiDAR, Convex hull, clustering, RANSAC*

1 Introduction

The precise representation of the driving environment is of utmost importance to the perception system of autonomous vehicles, which is considered a primary function to facilitate the safe operation of autonomous vehicles in complex driving scenarios [1].

Received 22 January 2026; Accepted 15 March 2026

Failure to address this function of the perception system of autonomous vehicles can result in serious accidents, as highlighted by several accidents involving autonomous vehicles, such as those involving Tesla vehicles [2] and Uber vehicles [3]. These accidents reveal the critical failure of the subject vehicles' perception systems to effectively and timely identify and classify key objects of interest along the vehicles' paths. Over the last few decades, substantial progress has been made in the development of autonomous driving. For autonomous vehicles to be considered safe, a reliable perception system that effectively perceives the environment for mobile platforms is a requirement. In the development of perception technology, a key area of focus is the development of 3D object detection. Many researchers have contributed significantly by exploring and implementing different techniques for the development of this research area [4-5]. In recent times, the use of 3D light detection and ranging (LiDAR) sensor technology has gained more recognition for its ability to achieve precision up to the centimeter level, even under all weather conditions, and for the ability of the sensor technology to achieve measurements over a long range, compared to the use of stereo cameras [6]. This research article aims to explore the research area of 3D object detection techniques using LiDAR technology for the precise localization of vehicle bounding boxes. LiDAR technology uses laser beams, which are directed towards the target, and the laser beam bounces back, resulting in the collection of 3D points. The processing of 3D points is an essential part of the research area. Among the different LiDAR sensor technologies available, the Velodyne LiDAR sensor has gained more recognition for its efficiency, especially for the development of Autonomous Driving, as it offers a 360-degree view of the scene.To conduct the research, it is essential to acknowledge that the data is acquired in the form of 3D scans, represented by point clouds. Regarding the bounding box, the significance of a 3D bounding box extends beyond image scenes, as it provides measurements of length, height, and depth in the LiDAR image. The 3D bounding box encompasses eight corners and edges, capturing the image on a plane defined by the x, y, and z coordinates. The point cloud extracted from LiDAR requires preprocessing to achieve the desired results as processing millions of points alone may not suffice. Therefore, several preprocessing steps are necessary, followed by segmentation to cluster objects and determine the appropriate approach for forming a bounding box. Visualizing the bounding box in a color image is crucial as it enhances human interpretation. This visualization can be verified by overlaying the bounding box on the image using calibration values.

In the field of autonomous driving systems, the task of 3D object detection usually requires the application of machine learning techniques. In order to test the efficiency of these techniques, certain established datasets, like KITTI [7], have been used. These datasets consist of a variety of images obtained through frontal cameras and lidars, and ground-truth information represented in the form of 3D boxes. While frontal cameras provide certain information regarding textures, which can be used for object classification, lidars and depth cameras provide information regarding depth, which helps in the estimation of object pose [8]. Most of the techniques have been based on the application of data fusion techniques, which can help in overcoming the limitations of certain types of sensors.

The domain of 3D object detection using LiDAR sensors can be broadly classified into two categories: traditional methods and deep learning methods. In recent times, several techniques using machine learning and deep learning methods have been introduced to offer end-to-end solutions to achieve outstanding performance in object detection. Nevertheless, these learning-based techniques have two major limitations. First, they are

highly dependent on manually labeled data, which is a tedious and time-consuming process. Second, these techniques are likely to face difficulties if the scene or setup is changed. In contrast, traditional methods are more flexible. The traditional pipeline of 3D object detection using LiDAR sensors usually starts with ground segmentation, followed by clustering of non-ground points, and then bounding box estimation of the object based on clustering results. The major advantage of traditional object detection methods is that they don't need prior information about the environment to train the model. This makes traditional methods have better generalization performance compared to learning-based methods [12]. In the proposed work, a methodology is proposed, which is based on traditional convex hull.

This research paper primarily focuses on the estimation of the vehicle's bounding box by clustering cloud points and convex hull in LiDAR images. The key contributions of this study can be summarized as follows:

- Employed the RANSAC algorithm to ensure the effectively handle outliers and prevent them from affecting the final bounding box estimation.
- The cloud points are represented in the KD- tree form and applied the clustering algorithm to cluster the cloud data points based on similarity.
- Estimate convex hull for area of interest and localize the vehicle using minimum area rectangle.

2 Related Work

The advancement of autonomous driving relies on several factors, including perception, positioning, scenario analysis, decision-making, and command control. Among these, the perception sub-system holds utmost significance as it determines the behavior of autonomous vehicles (AVs) based on their surroundings. Additionally, the safety of road users sharing the same space with AVs heavily relies on the AVs' ability to detect and identify objects. As a result, object detection has emerged as a crucial research domain in the perception of self-driving vehicles [1].

In the world of automation today, it is crucial to take into account the possibility of collisions that can occur during actions such as lane changing and overtaking. According to Salvatore Cafiso et al. [2], there have been suggestions regarding the development of methods to minimize collisions among different vehicles. One way of testing the reduction of collisions is through the use of Monte Carlo simulation. Among the intelligent systems developed, the one which stands out is the Adaptive Cruise Control (ACC), which has shown a reduction in the number of collisions ranging from 3% to 25%. ACC is seen to be especially relevant for medium trucks because they move at a faster pace than heavy trucks. Thus, ACC is seen to be vital in the reduction of collisions and is going to be a key factor in the future.

The multimodal fusion framework proposed by the authors of [3] includes the processing of 3D LIDAR point cloud and RGB image information, leading to accurate vehicle position and size estimation in Bird's Eye View (BEV). For accurate semantic segmentation of RGB images, an efficient Convolutional Neural Network (CNN) structure named ERFNet is used. The lightweight CNN structure for semantic segmentation of projection-based LiDAR point clouds proposed in [4] was found to have a lightweight structure with only

1.9M parameters, resulting in an impressive 87% reduction in parameters compared to the existing state-of-the-art networks.

The authors of [4] proposed a novel approach for representation learning of LiDAR point clouds using siamese LocNets. The learned representations by LocNet were then utilized in a global localization framework that incorporated range-only observations. The effectiveness and accuracy of the localization system were evaluated on the KITTI dataset, demonstrating impressive performance.

Huang et al. [5] proposed an ongoing vision framework that uses the concept of "right shell tree" and "left shell tree." Huang et al. also proposed a rapid raised convex hull algorithm, based on a binary tree, for scattered points. This algorithm can be effectively used for reducing the size of scattered points by removing non-convex hull vertices while identifying the vertices of the convex frame. This proposed algorithm can simplify the decision-making process for the associated connection of the vertices of the convex hull.

M.Himmelsbach et al., in their research paper [6], proposed a method for 3D LiDAR segmentation. In the paper, the author proposed a new method for the segmentation of planes by dividing them into angular segments, as shown in the paper. In the proposed method, the data acquisition and pre-processing were performed by taking into consideration the frames acquired by the real sensor and dividing them accordingly. For the local ground plane, the iterative model was used for both sloped and flat roads. In the proposed method, the segmentation was performed by the voxel grid method.

Similarly, Klaas Klasing et al. [7] proposed a clustering-based approach for effective segmentation. They compared and evaluated the RBNN (radially biased Nearest Neighbor) and KNN (k Nearest Neighbor) clustering methods to determine their relative performance.

Several approaches have been employed for encoding 3D point clouds in classification tasks. One common method involves representing the point cloud using a voxel grid and utilizing feature detectors. In [8], Rusu et al. introduced the Point Feature Histograms (PFH) and Viewpoint Feature Histograms (VFH) that leverage the spatial relationships among neighboring points to calculate features and generate a descriptor. Aiswarya G et al. [9] propose one reason for curiosity in content-based image recovery, such as obtaining comparable 3D images by inputting a 3D picture. Content-based picture recovery considers features like texture and shape of the input image during the retrieval process. This study introduces a technique for recovering substance-based 3D images using the point cloud library (PCL), an open project for processing 2D/3D images and point clouds. The process involves four key steps: key-point identification, feature extraction, correlation, and visualization, with the ultimate outcome being the retrieval of a collection of 3D images closely related to the input image.

Several deep learning approaches have been employed for bounding box localization of vehicle in LiDAR images. Some of such approaches are discussed here. DeepMultiBox VeloFCN [10], FVNet[11] and PointRCNN[12]are CNN based approach for bounding box localization. Table 1 summarizes the various state-of-the-art techniques employed for bounding box localization.

Table 1: State-of-techniques for bounding box localization in LiDAR images

Method	**Advantages**	**Limitation**
FVNet[11]	Front-View Proposal	Computational complexity

	Generation	
PointRCNN[12]	End-to-End Learning and Multi-View Feature Fusion	Computational complexity and Limited Generalization
VoxelNet [13]	End-to-End Learning and Accurate 3D Object Detection	May not capture fine-grained details of objects, leading to potential limitations in detecting and localizing small or densely packed objects.
BirdNet [14]	region proposals from bird's-eye-view	Weak result
RT3D [15]	region proposals from bird's-eye-view	Weak result
LMNet [16]	front-view representation	unsatisfactory results because of the loss of details
Deep MANTA [17]	Coarse-to-Fine Approach, Many-Task Learning	Reliance on Monocular Images and computational complexity

The literature of this research discussed various approaches for bounding box localization in LiDAR images and also analyzed the deep learning approaches that are more computationally expensive for bounding box detection. The current work proposed a method using mean shift clustering-based approach for bounding box localization.

3 The Proposed Method

This section presents the workflow of the proposed methodology and provides a detailed description of the algorithms employed to achieve the desired results. Figure 1 depicts the schematic flow diagram to obtain bounding box both on LiDAR and camera image. This work employed Kitti dataset. The KITTI dataset is a widely used benchmark dataset in computer vision and autonomous driving research. It was developed through a collaboration between Toyota Technological Institute and the Karlsruhe Institute of Technology (KIT). The dataset provides real-world data collected from sensors such as cameras, LiDAR, and GPS, enabling tasks like object detection, tracking, and depth estimation. It is publicly available and primarily intended for academic and non-commercial research purposes. The arrangement to capture dataset consists of camera and LiDAR sensors both mounted on vehicle, to capture values, it can be performed by placing 4 cameras. Two gray and two-color cameras are placed on either side along with a velodyne laser sensor. A vehicle with sensors and camera captures 100k points per cycle with 10 frames per second. 3D LiDAR data is collected in the form of x,y.,z, values. These parameter values provide information about 3D scenarios. Further processing of points is required to provide the expected outcome. The process is carried out by the following methods:

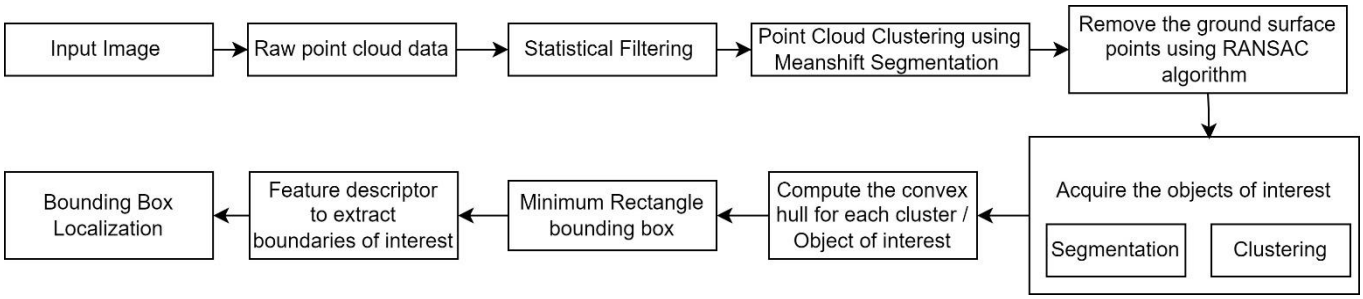


Fig 1. Flow of Proposed Methodology

3.1 Pre-Processing

The points collected from LiDAR sensors accumulate noise and outliers. Here, noise or outliers are the points which exceed threshold value defined. To remove these unwanted points pre-processing is carried out by applying Statistical filtering and ransac algorithm.

Pre-processing aims at removing outliers which usually include random errors collected while capturing data and it also removes the ground surface of roads to obtain objects of interest in lidar frames.

Statistical Filtering: In this algorithm the raw data points collected from sensor are processed by applying Meanshift and Standard deviation. Meanshift applies the point iteratively towards the dense region of points. These points in dense region are defined with certain threshold through which centroid is detected, and the points will assign to next dense region. This process is carried out until the maximum average distance between dense region of points is calculated with t times of standard deviation. If the distance between the points exceeds the standard deviation value, then it is considered as outliers and rest as inliers. Figure 2 depicts the process involved in filtering process.

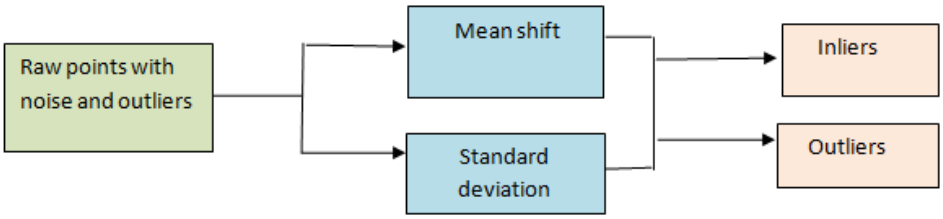


Fig 2. Filtering process

RANSAC: Random sample consensus is an iterative process to remove the ground surface points which are present on road to obtain the target cluster for further process. The utilization of the RANSAC algorithm is driven by its robustness as an estimation technique, allowing it to effectively handle outliers by iteratively fitting models to subsets of data and identifying the best-fitting model. This iterative process enables RANSAC to determine the optimal parameters representing the bounding box. Additionally, the algorithm is known for its fast and efficient performance.

RANSAC algorithm is initiated by applying two conditions, that are:

- Threshold should set manually starting from minimum values.
- Model must be defined as a priority, with respect to data points.

ALGORITHM I: RANSAC ALGORITHM

Input: Point cloud data (PointCloud)
Parameters:
- MaxIterations: Maximum number of iterations
- Threshold: Distance threshold to classify points as ground/non-ground
- SampleSize: Number of points to randomly sample for model estimation

Output: Filtered point cloud data without ground points (Inlier points)
Steps:

1. Initialize an empty set to store the inlier ground points.
2. Repeat for MaxIterations:
   a. Randomly sample SampleSize points from the point cloud.
   b. Fit a model (e.g., plane) using the sampled points.
   c. Calculate the distance between all points and the estimated model.
   d. Count the number of inliers (points within the threshold distance to the model).
   e. If the number of inliers is above a predefined threshold:
      - Refit the model using all inliers.
      - Add the inliers to the set of ground points.
3. Remove the ground points from the original point cloud data.
4. Return the filtered point cloud data

The process mentioned in Algorithm I is carried out iteratively until all the points are covered or maximum iteration is reached. After applying the RANSAC algorithm, the LiDAR data is obtained with the ground surface effectively removed. To further minimize computational complexity and processing time, the remaining points are segmented into smaller, structurally distinct clusters representing objects of interest.

3.2 Clustering and Segmentation

After pre-processing, the data collected points are used for further analysis. To isolate the objects of vehicles in this context—from the LiDAR data, clustering and segmentation techniques are employed.
Segmentation is a process of applying points into group based on similarities and cluster is a process of finding similarity in points so that they can be grouped. A simple Euclidean cluster extraction is implemented for the clustering process. Segmentation is performed using 3D grid subdivision of space, this space will use data structure algorithm. Here, K-D Tree is an algorithm used for segmentation.
K-D tree is one among the data structures to organize points in space with k dimensions. The basic idea :

- Spaces are divided into cells.
- Object primitives are placed into cells.
- These object primitives in the same cell are checked for collision.
- Pairs of primitives that do not share the same cells are not tested.

The algorithm is explained as given in Algorithm II.

ALGORITHM II: CLUSTER FORMATION ALGORITHM

Input: Set of points (P) obtained from computing KD Tree
Output: Points belonging to cluster C

1. Create a KDTree representation for the input point cloud dataset.

2. Initialize an empty list of clusters (C) and an empty queue (Q) for points.

3. For each point (pi) in P, do the following:
- Add pi to the current queue (Q).
- While Q is not empty, do the following for each point (pk) in Q:
- Search for a set of points (pk) that are neighbors of pi within a specified radius.
- For each point (pk) in the set, check if it has already been computed. If not, add it to the queue (Q).
- Once all points in Q have been computed, add Q to the list of clusters (C) and empty the queue (Q).

4. Repeat step 3 until all points (p) in P have been processed and moved to a cluster in C.

A K-D tree is employed to efficiently structure points in a k-dimensional space. It organizes the data in the form of a binary search tree, where each node divides the space along a selected dimension using a hyperplane perpendicular to that axis. Points with smaller values than the root are placed on the left, while larger values are positioned on the right, enabling efficient nearest-neighbour searches.
Following this, the points are grouped according to their similarity. Cluster extraction is then performed by evaluating the Euclidean distance between points, allowing similar points to be identified and grouped together.

3.3 Convex Hull

It is a shape extractor. The cluster object points extracted from the previous method may not always be recorded properly. To address this, a shape extractor algorithm is applied Convex hull acts as shape extractor which completely encloses the collection of points with minimum number of nodes.
Two main properties of convex hull are:
1. All the points in convex hull must always be facing any point intended towards inward, if it is concave referring even single point outward it has to exclude the point. By this the overall perimeter count value will be reduced.

2. The most extreme point of an axis is part of convex hull.

The flow diagram convex hull is given in Figure. 3. Figure 3 shows the flow chart of convex hull which has to deal with minimum number of points. It will enclose all minimum boundary points.

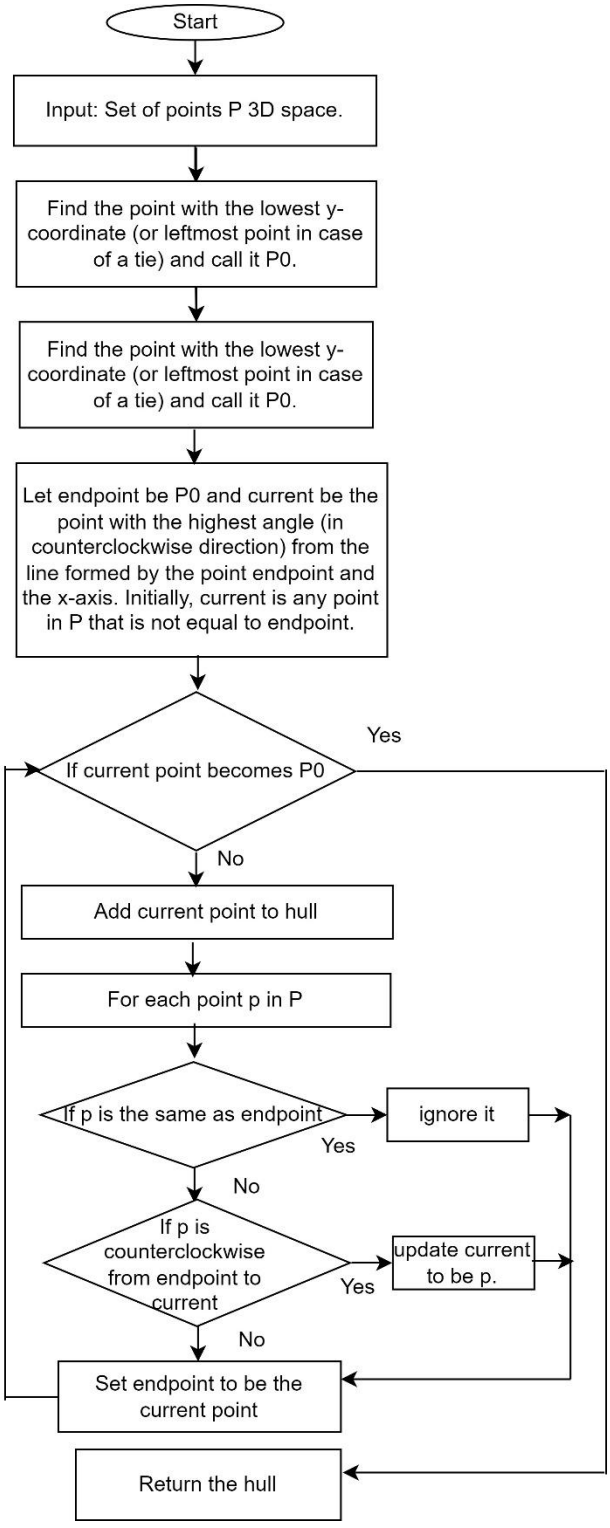


Fig 3. Flow diagram of convex hull

3.4 Minimum Area Rectangle

This algorithm is applied on the clustered objects obtained from previous step. The initial Bounding Box is drawn by preserving the height of objects since it takes maximum and minimum value points into consideration. This method retains 3D orientation box. It is applicable to all well-defined cluster points except for the few obstruct and point values

which are not recorded properly. Steps involved to determine minimum area rectangle is outlined in Algorithm III.

ALGORITHM III: MINIMUM AREA RECTANGLE

Input: Clustered objects obtained from the previous step
Output: Minimum area rectangles for each object

1. Iterate over each clustered object:
   a. Compute the minimum and maximum points along the height axis.
   b. Initialize the initial bounding box using the minimum and maximum points.
   c. Set the initial minimum area of the bounding box to a very large value.
   d. Set the initial rotation angle of the bounding box to 0.

2. Iterate over each clustered object:
   a. Generate all possible rotations of the bounding box.
   b. For each rotation:
      i. Calculate the area of the rotated bounding box.
      ii. If the calculated area is smaller than the current minimum area:
         - Update the minimum area to the calculated area.
         - Update the rotation angle of the bounding box.

3. Return the final minimum area rectangles for each object.

3.5 Feature Descriptor

A Descriptor is used to detect the interest points obtained from clusters. Movement of Inertia (MOI) and eccentricity-based descriptors are used in proposed work. Eccentricity may sometimes act as shape extractor. It is the ratio of major axis length to the minor axis length. The MOI is used with the bounding box; the value of bounding box is known but the rotation of box varies with displacement of object. So, MOI helps in retaining the bounding box around the object even when there is displacement.

3.6 Calibration Data

Camera Calibration generates data models which include intrinsic parameter and extrinsic parameter. Each data model includes Intrinsic Parameter such as focal length, skew metric, distortion and image centre. Extrinsic Parameter contain Position and orientation.

The terms of intrinsic parameters are:

Focal length: Focal Length is denoted as f. The distance between image plane and optic centre is said to be focal length.

Principle Point: The point in which the intersection between the image plane and the principal axis occurs. The principal point is denoted by u and v.

Skew Co-efficient: When sensor is placed the coefficient angle between x and y has to be perfectly rectangle, minute change in angle may lead to shape of parallelogram which in turn leads to skew coefficient

Distortion Co-efficient: It is a tangential displacement of lens; it occurs when lens is not parallel to the image plane.

$$Intrinsic\ Paramter\ K = \begin{bmatrix} f_x & S & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

Extrinsic Parameter includes R and T which is Rotation and Translation given by,

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \quad (2)$$

World coordinates relation: By Minimum Area Rectangle it resulted in 3D coordinates of x,y,z on each corner of Bounding Box which encloses an object. To apply Bounding box around two-dimensional image the equation is given by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = PT_2 \; R_1 \, T_1 \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3)$$

Where, T_2 = Translation from LiDAR to Velodyne camera.
T_1 = Translation from Camera-to-Camera Values.
P = K$[R \quad t]$
Where R= Rotation.
K = Intrinsic Parameter.

The transformation matrices T_1 and T_2 represent rigid body transformations between coordinate frames, namely camera-to-camera and LiDAR-to-camera, respectively. These transformations are expressed in homogeneous coordinates as:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (4)$$

where $R \in \mathbb{R}^{3\times3}$denotes the rotation matrix and $t \in \mathbb{R}^{3\times1}$represents the translation vector. Since the rotation matrix $R$is orthogonal, it has full rank, i.e., rank$(R) = 3$. The augmented homogeneous transformation matrix includes an additional row $[0\ 0\ 0\ 1]$, which is linearly independent of the other rows. Therefore, all four rows of the matrix are linearly independent, leading to:

$$\text{rank}(T_1) = \text{rank}(T_2) = 4 \quad (5)$$

This confirms that both $T_1$and $T_2$are full-rank matrices, ensuring that no spatial information is lost during the coordinate transformation process.

Bounding box: Bounding Box on an image is formed using the above world coordinate equation (3). Since the image is in 2D it is necessary to obtain the result in two values for each corner point. Using this Bounding Box is formed around the respective target on image.

4 Result Analysis

This section presents the experimental results obtained from the proposed vehicle detection and localization approach using both LiDAR point clouds and RGB images from the KITTI dataset. The methodology, as outlined in previous sections, is validated through a sequence of preprocessing, clustering, shape analysis, and bounding box generation steps.

4.1 Dataset Description and Scene Visualization

The experiments are performed using the KITTI Vision Benchmark Suite, which includes real-world driving scenarios captured using synchronized RGB cameras and LiDAR scanners. They have developed a very huge platform for only academic purposes with the helping funds from Toyota technologies and Karlsruhe Institute of Technology.
A representative frame from the dataset is shown in Fig. 4, depicting a typical road scene under real-time conditions. The RGB images serve to visually correlate with the LiDAR-based detections.
LiDAR data is provided in the form of 3D point clouds comprising (x, y, z) coordinates, which are stored as text files and subsequently processed for visualization and computation.

Fig 4. A Frame from Kitti Dtaset in road scenario.

4.2 Ground Removal and Point Cloud Preprocessing

To simplify the detection process, a ground removal step is applied to the raw LiDAR point cloud. This preprocessing step employs statistical filtering and the RANSAC algorithm to remove planar ground points, thereby isolating potential obstacles and vehicles. Figure 5 and Figure 6 illustrate the LiDAR view before and after ground removal, respectively. The output clearly shows a clean separation of objects of interest (e.g., vehicles) from the surface of the ground.

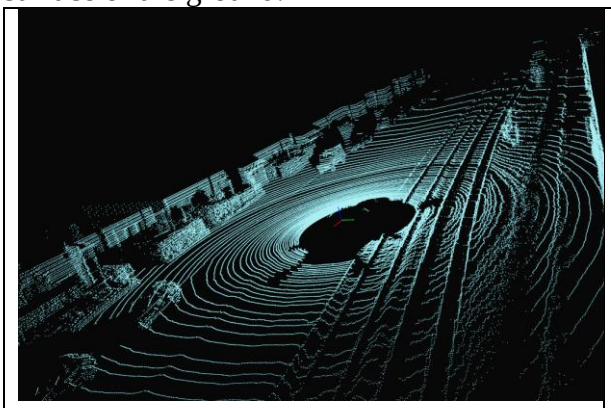


Fig 5. Different view of LiDAR image of same camera image.

Fig 6. After Removing of ground level

4.3 Clustering and Segmentation of Vehicles

After that, a clustering process is used to cluster the remaining points that are not on the ground. As depicted in Figure 7 and Figure 8, different clusters representing different vehicles are identified from the processed LiDAR points. These clusters are then used to extract the shapes and generate the bounding boxes.

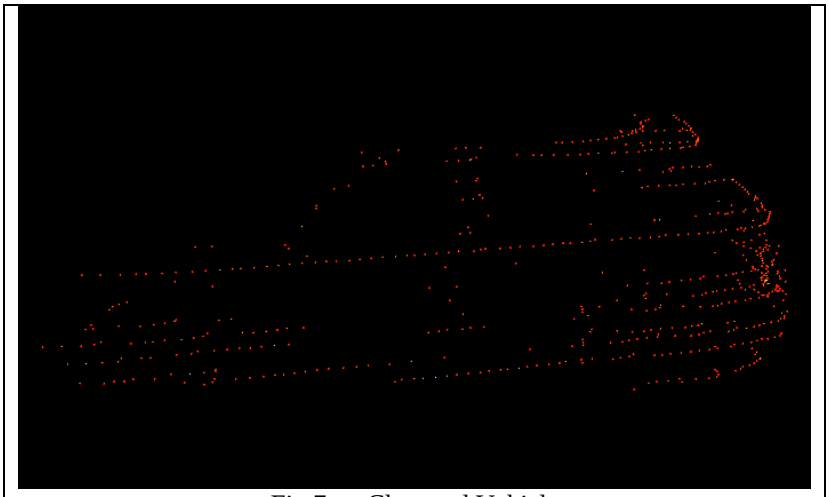


Fig 7. Clustered Vehicle

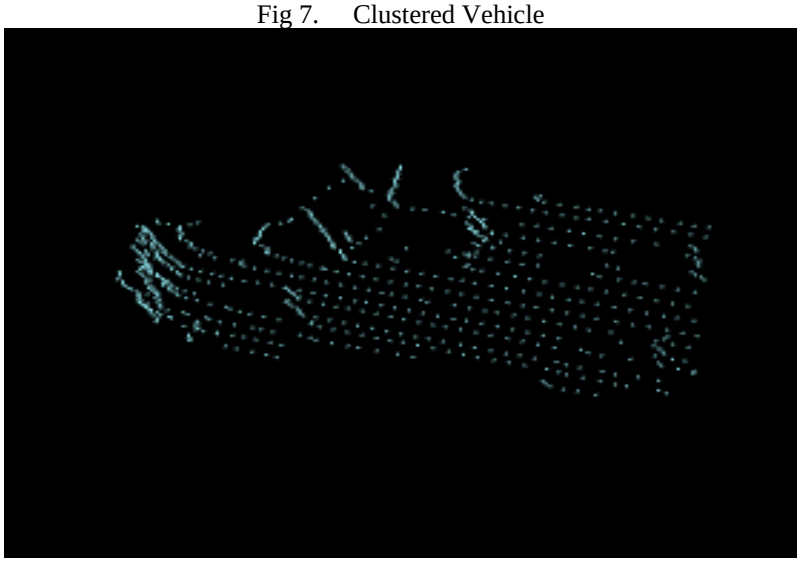
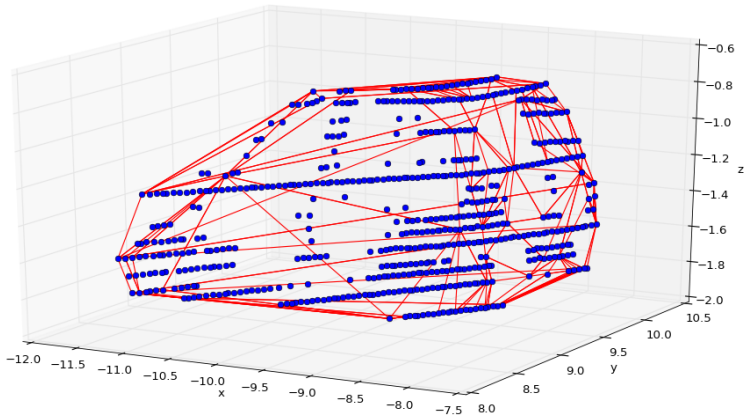


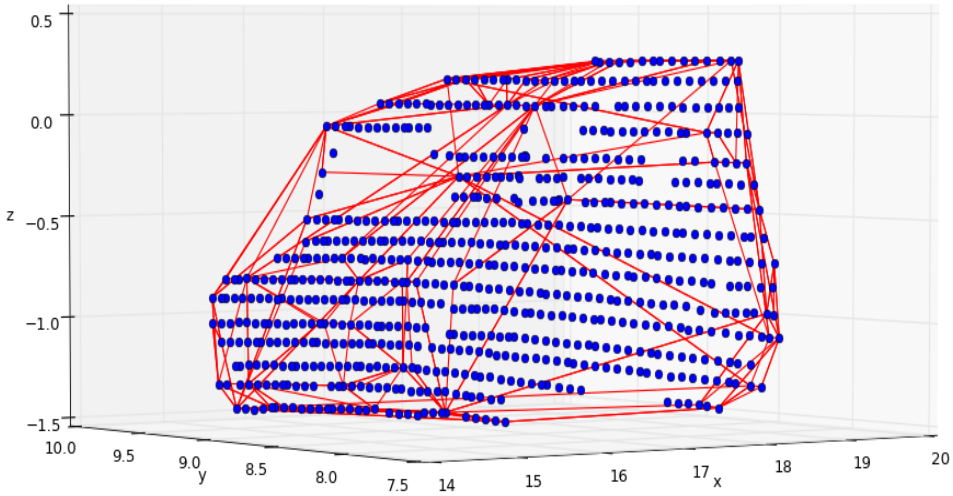
Fig 8. Clustered Vehicle of different view

4.4 Shape Extraction Using Convex Hull

To estimate the shape and size of the detected objects, a convex hull is computed on the clustered segments. Figure 9(a) shows the convex hull computed on the image of a car, while Figure 9(b) shows the convex hull computed on the image of a jeep. Though the convex hull does not encapsulate the object tightly, it is a reliable estimate of the boundary, which is extremely important for the recognition/localization of objects.



(a)



(b)

Fig. 9: Convex hull representation of vehicles from LiDAR data
(a) Car with smoother geometry and fewer hull vertices
(b) Jeep/SUV with box-like structure and higher number of hull vertices

The variation in the number of points in the convex hull among different vehicles results from the variation in their geometric structure. For a given set of points \(S ⊂ R^3\), the convex hull is given by the following equation 6:

$$\text{Conv}(\boldsymbol{S}) = \left\{ \sum_{i=1}^{n} \boldsymbol{\lambda_i p_i} \mid \boldsymbol{p_i} \in \boldsymbol{S}, \boldsymbol{\lambda_i} \geq \mathbf{0}, \sum_{i=1}^{n} \boldsymbol{\lambda_i} = \mathbf{1} \right\} \quad (6)$$

where the vertices correspond to the extreme boundary points. The number of such vertices $|\boldsymbol{V}|$ depends on the number of LiDAR points $\boldsymbol{N}$ and the geometric complexity of the object:

$$|V| \propto \kappa(S)\, N \quad (7)$$

where $\boldsymbol{\kappa}(\boldsymbol{S})$ represents a shape-dependent complexity factor.
As shown in Figure 9(a) and Figure 9(b), a clear distinction is observed between different geometries of vehicles. Figure 9(a) shows a car, which has smoother and more curved surfaces, hence a lower number of extreme boundary points and a lower number of convex hull vertices. Figure 9(b) shows a jeep, which is of SUV type and has a more angular and box-like shape, hence a higher number of sharp edges and surfaces, resulting in a larger number of extreme boundary points and a larger convex hull.

4.5 Bounding Box Generation in LiDAR and RGB Domains

Figure 10 illustrates the process of generating bounding boxes for the identified vehicles in the LiDAR domain. These bounding boxes are created based on the data points obtained through the cluster and shape extraction methods, and they form the final output of the proposed detection system.

For the detection output obtained through the LiDAR domain to be mapped onto the appropriate RGB images, the calibration data for the cameras and LiDAR sensors are

utilized. Figure 11 illustrates the RGB images for the detection output obtained through the LiDAR domain, where bounding boxes have been projected onto the 2D RGB images. It is worth noting that the LiDAR sensor has a higher ability to detect vehicles from a larger distance compared to the RGB images, which validates the superiority of LiDAR sensors for the perception of the environment.

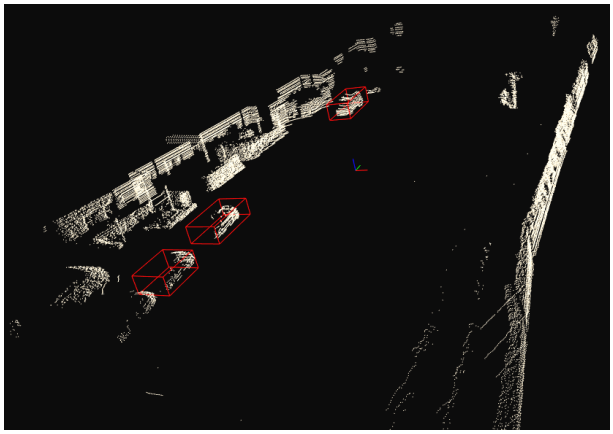


Fig. 10 Bounding Box around vehicles on LiDAR

Fig. 11 Bounding Box in color image

4.6 Evaluation Against Ground Truth

The KITTI dataset has ground truth bounding boxes for specific frames of the video. These are used for the qualitative evaluation of the precision of the proposed method. By comparing the bounding boxes obtained through the proposed method and the actual bounding boxes, the errors that have occurred during the estimation are noted. Figure 12 illustrates a comparative diagram showing the variation between the actual and estimated bounding box size for a number of frames of a single vehicle. In the proposed method, the Mean Absolute Error (MAE) was 0.167, the Root Mean Squared Error (RMSE) was 0.186, and the average percentage error was about 8.5%. These values prove the precision and reliability of the proposed bounding box generation method for various types of road scenarios.

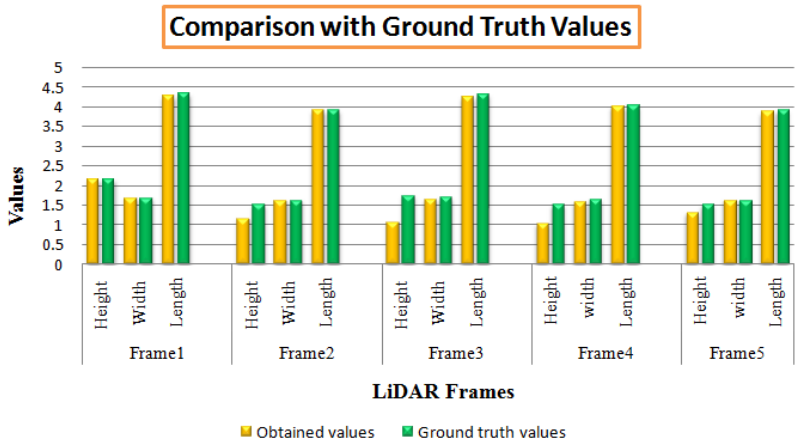


Fig 12. Comparison graph

In order to further test the accuracy of the proposed method for determining the bounding boxes, a residual error test was performed to calculate the differences between the results obtained and the known values for height, width, and length using five different LiDAR images. The results are shown in Figure 13.

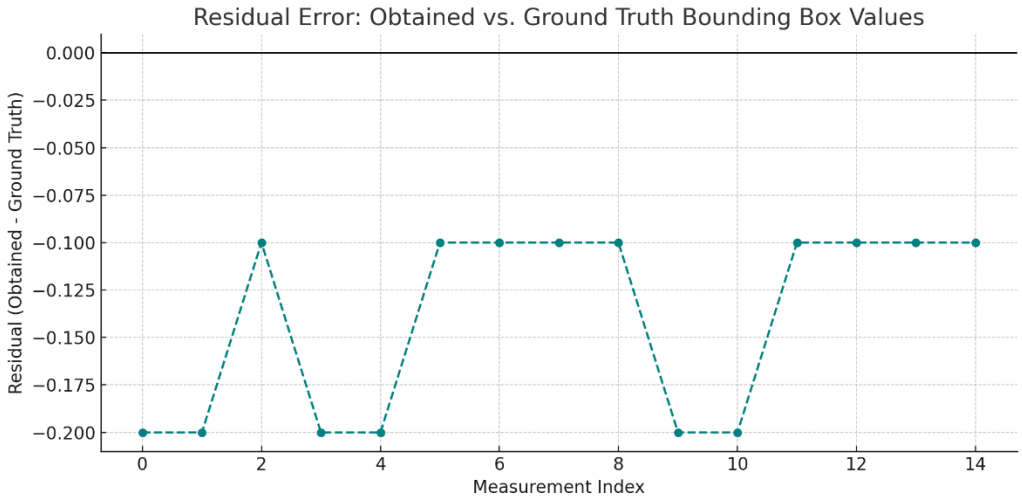


Fig. 13 : Residual error plot for the obtained dimensions of the bounding box compared to the ground truth values for the five frames of the LiDAR sensor.

The negative values in Figure 13 represent a slight underestimation of the dimensions of the objects, which is consistent. From the above Figure 13 , it can be observed that the residual value is consistently negative, and it varies between -0.1 and -0.2 in most cases. This indicates that the bounding box size estimated using the proposed method is slightly less compared to the actual size, which is a characteristic of underestimation. The magnitude of the residual is low and consistent in all frames, indicating a consistent model behavior.Such consistent underestimation may result from:

- Conservative scaling during convex hull estimation,
- Fixed thresholds in the clustering process,
- Rounding or projection limitations during LiDAR-to-RGB calibration.

However, the residuals are still within a reasonable range, especially for autonomous purposes where underestimation is better than overestimation.

This analysis proves the statement that the method offers a reliable and understandable estimation of the object's dimension with minimal deviation from the ground truth, validating the robustness of the proposed approach.

For the evaluation of the spatial overlap between the predicted values and the actual values of the bounding box, a one-dimensional Intersection over Union (1D IoU) metric was computed for each dimension, i.e., height, width, and length, for five different LiDAR frames. The IoU values varied between 0.889 and 0.978, and the average IoU was 0.936, indicating a high degree of alignment between the values. It was also observed that the length dimension showed the highest IoU, indicating a high degree of precision for the longitudinal axis of the vehicle. This also validates the precision of the proposed geometric method for spatial fidelity.

5 Conclusion

To conclude, in this work our main goal is to detect the vehicle and localize it by generating the bounding box around the target on LiDAR and RGB image. Also, the algorithm applied is fine suited for any road situation either flat, slope and irregular and so on. Considering the aspect of fully automated car suggested having sensor set, which has far vision like LiDAR as it covers free space compared to camera and purpose of bounding box is to estimate the distance or space coverage of that vehicle with proper measurements with length, height and depth.

In future work, the technique can be used for tracking at different stages like stagnant, dynamic or in motion and also for lane changing as tasks of autonomous vehicles.

References

[1]. Eduardo Arnold, Mehrdad Dianati, Robert de Temple and Saber Fallah, "Cooperative Perception for 3D Object Detection in Driving Scenarios using Infrastructure Sensors", arXiv:1912.12147v2 [cs.CV] 30 Oct 2020

[2]. Technical report, Tesla Crash," National Highway Traffic Safety Administration, US Department of Transportation, Tech. Rep. PE 16-007, January 2017.

[3]. "Preliminary report, Highway HWY18MH010," National Transportation Safety Board, US Government, Tech. Rep. HWY18MH010, May 2018

[4]. Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," arXiv preprint arXiv:2206.10092, 2022

[5]. X. Ma, W. Ouyang, A. Simonelli, and E. Ricci, "3d object detection from images for autonomous driving: a survey," arXiv preprint arXiv:2202.02980, 2022

[6]. J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9297–9307, 2019

[7]. A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012, pp. 3354–3361

[8]. E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A Survey on 3D Object Detection Methods for Autonomous Driving Applications," IEEE Transactions on Intelligent Transportation Systems, 2019.

[9]. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[10]. J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," IROS, 2018.

[11]. C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection From RGB-D Data," in 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018

[12]. Ningning Ding, Ruihao Ming, Bo Wang, Yafei Gu, "An Efficient Convex Hull-Based Vehicle Pose Estimation Method for 3D LiDAR", arXiv:2302.01034v1 [cs.CV] 2 Feb 2023

[13]. Pascal Housam Salmane, Josué Manuel Rivera Velázquez, Louahdi Khoudour, Nguyen Anh Minh Mai, Pierre Duthon , Alain Crouzil , Guillaume Saint Pierre and Sergio A. Velastin, "3D Object Detection for Self-Driving Cars Using Video and LiDAR: An Ablation Study", Sensors, Vol. 23, No. 3223, 2023, pp. 1-20, https:// doi.org/10.3390/s2306322

[14]. Salvatore Cafiso and Alessandro Di Graziano, "Evaluation of the effectiveness of ADAS in reducing multi vehicle collisions", International Journal of Heavy Vehicle Systems, Vol .19, No.2, May 2012.

[15]. R. Barea et al., "Vehicle Detection and Localization using 3D LIDAR Point Cloud and Image Semantic Segmentation," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 2018, pp. 3481-3486, doi: 10.1109/ITSC.2018.8569962

[16]. Xie X, Bai L, Huang X. "Real-Time LiDAR Point Cloud Semantic Segmentation for Autonomous Driving. Electronics", Vol. 11, No. 11, 2022, pp. 1-13, https://doi.org/10.3390/electronics11010011

[17]. Huang, Linna, and Guangzhong Liu. "Proved quick convex hull algorithm for scattered Points," International Conference on Computer Science and Information Processing, pp:1365-1368. Aug- 2012.

[18]. M. Himmelsbach, Felix v. Hundelshausen and H.-J. Wuensche, "Fast Segmentation of 3D Point Clouds for Ground Vehicles", Intelligent Vehicles Symposium University of California, pp :21-24, June 2010.

[19]. Klasing, Klaas, Dirk Wollherr, and Martin Buss. "A clustering method for efficient segmentation of 3D laser data," International Conference on Robotics and Automation, pp: 4043-4048. May- 2008.

[20]. R. B. Rusu, N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 3212-3217, doi: 10.1109/ROBOT.2009.5152473.

[21]. Aiswarya, G., Valsaraj, N., Vaishak, M., Nair, T. U., Karthik, V., & Nair, J. J, "Content- based 3D image retrieval using point cloud library a novel approach for the retrieval of 3D images," International Conference on Communication and Signal Processing, 2017 pp: 0817-0820

[22]. B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," in Robotics: Science and Systems, 2016.

[23]. Jie Zhou, Xin Tan, Zhiwen Shao and Lizhuang Ma, "FVNet: 3D Front-View Proposal Generation for Real-Time Object Detection from Point Clouds", arXiv:1903.10750v3 [cs.CV] 26 Nov 2019, pp. 1-8

[24]. Shaoshuai Shi, Xiaogang Wang, Hongsheng Li, "PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud", arXiv:1812.04244v2 [cs.CV] 16 May 2019, pp. 1-10

[25]. Yin Zhou, Oncel Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection", arXiv:1711.06396v1 [cs.CV] 17 Nov 2017, pp. 1-10

[26]. J. Beltran, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: A 3d object detection framework from lidar information," in IEEE International Conference on Intelligent Transportation Systems. IEEE, 2018, pp. 3517–3523

[27]. Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3434–3440, 2018

[28]. K. Minemura, H. Liau, A. Monrroy, and S. Kato, "Lmnet: Real-time multiclass object detection on cpu using 3d lidar," in Asia-Pacific Conference on Intelligent Robot Systems. IEEE, 2018, pp. 28–34.

[29]. Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Celine Teuliere, Thierry Chateau, "Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image", arXiv:1703.07570v1 [cs.CV] 22 Mar 2017