Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Automobile indexation from 3D point clouds of urban scenarios

Ramirez-Pedraza Alfonso, González-Barbosa José-Joel, Ramirez-Pedraza Raymundo, González-Barbosa Erick-Alejandro & Hurtado-Ramos Juan-Bautista

To cite this article: Ramirez-Pedraza Alfonso, González-Barbosa José-Joel, Ramirez-Pedraza Raymundo, González-Barbosa Erick-Alejandro & Hurtado-Ramos Juan-Bautista (2021) Automobile indexation from 3D point clouds of urban scenarios, Automatika, 62:3-4, 311-318, DOI: 10.1080/00051144.2021.1947609

To link to this article: https://doi.org/10.1080/00051144.2021.1947609

Published online: 01 Jul 2021.

AUTOMATIKA
2021, VOL. 62, NOS. 3–4, 311–318
https://doi.org/10.1080/00051144.2021.1947609

REGULAR PAPER

OPEN ACCESS Check for updates

Automobile indexation from 3D point clouds of urban scenarios

Ramirez-Pedraza Alfonso [a], González-Barbosa José-Joel [b], Ramirez-Pedraza Raymundo[c], González-Barbosa Erick-Alejandro[d] and Hurtado-Ramos Juan-Bautista [b]

[a]Cátedra CONACYT – Centro de Investigaciones en Óptica A.C., León, Guanajuato, México; [b]Instituto Politécnico Nacional, CICATA-Qro, Querétaro, México; [c]CINVESTAV, Zapopan, Jalisco, México; [d]Tecnológico Nacional de México/ITS de Irapuato, Irapuato, Guanajuato, México

ABSTRACT

In this paper, we introduce a methodology for the detection and segmentation of automobiles in urban scenarios. We use the LiDAR Velodyne HDL-64E to scan the surroundings. The method is comprised of three steps: (1) remove facades, ground plan, and unstructured objects, (2) smoothing data using robust principal component analysis (RPCA), and finally, (3) unstructured objects model and indexing. The dataset is partitioned into training with 4500 objects and test with 3000 objects. Mean Shift thresholds, the filter, the Delaunay parameters, and the histogram modelling are optimized via ROC analysis. It is observed that the car scan quality affects our method to a lesser degree when compared with state-of-the-art methods.

ARTICLE HISTORY
Received 3 September 2020
Accepted 14 June 2021

KEYWORDS
Automobile indexation; 3D points cloud; segmentation; indexing

Introduction

Object recognition from point clouds is a challenging computer vision problem due to noise, sparse data, and scenarios' wide variability. Moreover, data acquired by the LiDAR Velodyne 64E sensors contains partially scanned objects, making the problem more interesting, and a common practice is to register multiple point clouds [1]. This paper proposes a method for the segmentation and recognition of automobiles in LiDAR generated point clouds. Figure 1(a) shows our acquisition platform capable of performing a mobile mapping from static or dynamically. Platform consist of three data sources: a LiDAR, a panoramic camera, and a Global Positioning System (GPS). Mobile mapping refers to the collection of data from multiple georeferenced sources. Applications are numerous, such as cartography, archaeology, geography, geomorphology, seismology, and atmospheric physics.

3D modelling of cities can solve traffic problems, prevent disasters in mines, and help design cities with organized growth [2]. For example, in [3], the authors perform 3D building detection and modelling by processing airborne LiDAR point clouds. As safety applications using airborne LiDAR scanning stand out from the work of [4], the authors' monitor power-line networks for vegetation clearance stating that the safety of the electrical network infrastructure can significantly affect our daily lives. In autonomous driving applications, the work presented in [5] stands out where they segment and classify objects from point clouds obtained with a LiDAR mounted on the roof of a vehicle. Their approach combines 2D and 3D techniques reaching real-time performance at 0.1 FPS. Object segmentation in 3D point clouds is a growing field of study due to the need to characterize and recognize objects scanned with LiDAR or segment sizeable 3D point clouds. Object segmentation is the early step towards more advanced robotic behaviours; for example, robots need to localize objects before attempting tasks such as grasping, manipulation, or path planning [6–11]. In [12] is proposed as a solution to automate mobile robots by segmenting the urban scene. In one point cloud, they stored the building's facade and the ground, and on another, they stored the foreground. Finally, the authors grouped objects such as cars, people, and walls.

Google's autonomous driving cars [13] can detect and track obstacles on their way for safe driving. The equipment of this car includes multiple sensors and cameras, including a LiDAR that generates a map of the environment; radars that detect the closeness of the objects allowing safe in traffic navigation; cameras located on the rear-view mirrors used to detect semaphore's lights; GPS, IMU (Inertial Measurement Unit) and an encoder on one wheel that determines the precise location of the car. Their system combines laser measurements with high-resolution maps to determine the location of the car. Our approach is quite different since we aim to segment parked or cars moving in urban scenes to perform 3D reconstruction of the scene without the cars leaving the rest of the scene objects such as facades, trees, lampposts, Etc. In [14], the authors propose a data fusion system based on scanner laser and computer vision. The pedestrians are detected using

CONTACT González-Barbosa José-Joel jgonzalezba@ipn.mx Instituto Politécnico Nacional, CICATA-Qro, Cerro Blanco 141, Col. Cimatario, CP 76090, Querétaro, México

a pattern matching approach with the LiDar data and Histogram of Oriented Gradients (HOG) with camera data. Both detections are fused, and the movement of the pedestrians is computed with Kalman Filter (KF), and Unscented Kalman Filter (UKF) approaches.

Regarding indexing and matching 3D point clouds, [15] introduced a recognition approach based on 3D interest points' indexing. A set of interest points represent each site, where each point contains a descriptor vector. A comparison between two sets of points decides the similarity between two places. In [16], the authors proposed a method to detect interest points in 3D meshes using a modified Harris detector. We use a different approach for the indexing using a histogram of normal vector directions to the object surface. With the 3D reconstruction of the urban environment without obstacles such as pedestrians, parked cars, Etc., we need to automatically detect and remove these obstacles. This work introduces a new method of segmentation, filtering, and detection of automobiles in point clouds.

1. Object segmentation

Point clouds of urban environments contain structured and unstructured objects. Structure objects are the ground and facades; unstructured objects do not have simple shapes such as trees, pedestrians, cars, Etc. Our segmentation method first detects and extracts structured objects from the point clouds, and then, the remaining points are segmented using Mean Shift. In [17], the authors provide a detailed assessment of the Mean Shift algorithm for the tree segmentation using airborne LiDAR data. Figure 1(b) shows the segmentation algorithm's modules and their place in our automobile indexing approach.

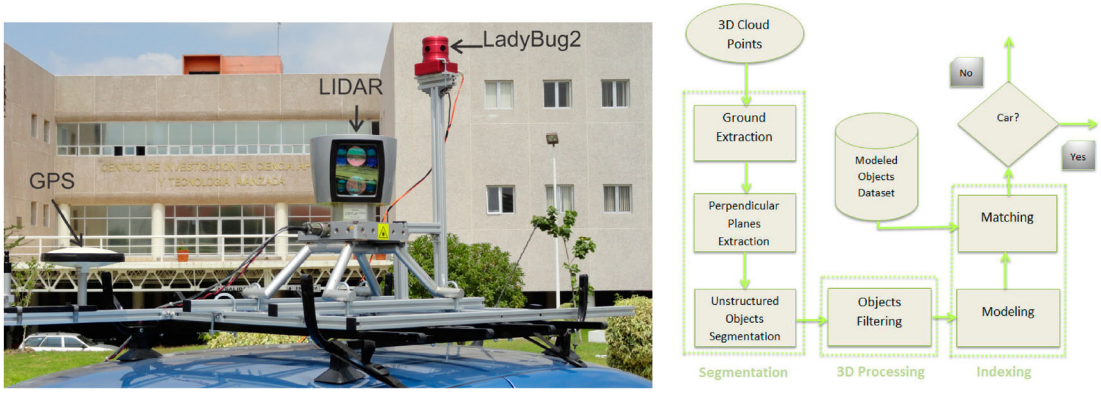
1.1. Planes extraction

Points from the ground plane are detected and extracted from the point cloud using the method proposed in [18], then we detect and extract the facades. The extraction of the points corresponding to the ground plane uses a threshold of α. In this work, we improve the threshold obtained from the measurement system's uncertainty using Equation (1) from [19]. It is common to use the expression (Equation (1)) to calculate an instrument's measurements; we adapted it to obtain a reference point that discriminates planes from other objects. We use the calibration obtained in [20], where the variance of the sensor calibration is 2.22 cm^2, and a mean error of 1.56 cm.

$$U = k * u_c(y) = k * \sqrt{U_{cal}^2 + U_p^2 + U_w^2} + |b| \quad (1)$$

Where U_{cal}^2 is the calibration variance of the LiDAR sensor, U_p^2 is the sum of the errors in the measurement process, U_w^2 is the average size of the sidewalk (13 cm), k is the coverage factor used to obtain a confidence level $p = 94.5\%$ in the uncertainty, and $|b|$ is the mean error. A threshold α is defined as $\alpha = U = 15.46$ cm.

We use the normal vector to the ground to define a new coordinate system for the points. Knowing that the facades are perpendicular planes to the ground, we did not use the third coordinate of the segmentation points. Then, using the modified Hough transform [18] we searched for the set of points that model a plane. The parametric space is given by ρ, θ, which are the normal vector parameters that pass through the origin on the modified Hough transform. Finally, we segmented the unstructured objects using MeanShift to obtain their location inside the point cloud. This algorithm groups a set of dimensions d, associating each point with the mode or peak of the data set's probability density function.



(a) (b)

Figure 1. (a) Acquisition Platform. Our system is mounted on the top of a car. The quality of the acquisition data allows us to drive in urban areas. The LadyBug camera captures 360° panoramic images, the GPS provides the location in latitude and longitude, and the LiDAR performs a 360° scan of the scene. (b) Procedure. Our method is composed of three stages: segmentation, filtering, and indexing. The segmentation stage extracts the ground, perpendicular planes to the ground, and unstructured objects. Filtering is applied to the objects to remove outliers. We apply an Indexing method to the filtered objects by obtaining a Delaunay triangulation and matching the normals' histogram against a library of models.

2. Filtering

We remove from the point cloud, the ground plane, and the facades; the remaining objects correspond to planes composed of few points and unstructured objects such as trees, pedestrians, cars, lampposts, Etc.

In this section, we describe the filtering step that we apply to the segmented objects. The filtering consists of removing points that affect the estimated normal vector's accuracy and lower the detection performance. Following an approach similar to [21], we estimated the normal vector in two steps: On the first step, we determined the neighbourhood size (r of the Equation (6)) for each point; on the second step, we correlated the estimation on edges and corners. To determine an appropriate neighbourhood size, we chose an initial r value and reduced it iteratively until the Equation (6) is true. Once the size is defined, we estimated a tangent plane to the neighbourhood and the normal. As a starting point, we obtained the k-nearest neighbours to each p_i. The next step is to adjust a plane to the surface using RPCA [22]. See Equations (2) and (3). For neighbour estimation, we used Mahalanobis distance (MD), which measures data dispersion concerning $\overline{p_w}$.

$$\overline{p_w} = \frac{\sum w_i p_i}{\sum w_i} \tag{2}$$

$$CM_w = \frac{1}{n-1}\sum_{i=1}^{n}(p_i - \overline{p}_w)(p_i - \overline{p}_w)^T W \tag{3}$$

where $W = \{\sqrt{w_1}, \ldots, \sqrt{w_n}\}$ are the weights associated to each point p_i in the neighbourhood. The filtered point is defined by :

$$p_i^f = ((p_i - p_i')^T * v_3) * v_3 + p_i' \tag{4}$$

where $p_i' = \overline{p_w} + v_1^T * (p_i - \overline{p_w}) * v_3$, $[U\ S\ V] = svd(CM_w)$, and $V = [v_1\ \ v_2\ \ v_3]$

The local curvature value is defined as

$$S = \sum_r \frac{2 * ((p_i - p_i')^T * v_3) * v_3}{r^2} \tag{5}$$

The following equation determines the distance of the neighbours defined for each point. This distance allows us to define small threshold distances when the curvature is large and large threshold distances when the curvature is slight.

$$|v_3^3 * (p_i - (\overline{p}_w + t_{\min}^T * v_3))| \leq \frac{S|[v_2 \quad v_3] * (p_i - \overline{p}_w)|}{2}$$

$$t_{\min} = \text{mean}\left\{\sum_{p_i \in \text{neig}(P)} v_3 * (|p_i - \overline{p}_w|)\right\} \tag{6}$$

where $p_i \in \text{neig}(P)$ corresponds to all points at a distance less than or equal to r from point p_i. Filtering points need to know the initial distance for search neighbours and the minimum number of neighbourhood points. It is required a minimum number of points in a neighbourhood to consider it an object; if not, it corresponds to a data noisily. These two parameters, the starting distance r of neighbours and the minimal number of points at a neighbourhood, are determined by a ROC analysis

3. Indexing

Different object instances or object classes often have other geometric shapes. Thus, a geometric descriptor uses an object's shape based on specific geometric features to index a particular instance of object or object class. In [23], the authors propose an approach that includes a novel formulation of a disparity term that simultaneously considers the structural similarity index. The indexation step uses the segmented and filtered objects and our dataset of object models. During indexing, the 3D objects are modelled and compared against the models in the dataset.

3.1. Modeling

3D models indexing and searching in a database is a process of coding and describing 3D models' shapes. The approach proposed in [24] classifies moving objects into four classes: vehicle, pedestrian, bicycle, and the crowd. The authors use LiDAR and camera data and modelling the information using four number-of-point-based features, eleven shape features, and nine statistical features. The orientation of the normal vector of the tangent plane on each point of the surface cab describes Objects' surfaces. In our proposed work, three steps perform the modelling :

(1) Orientation normalization. Each segmented object is composed of a set of points Z_i oriented to the data distribution; for this, we calculate the eigenvectors $V = [v_1, v_2, v_3]$ and eigenvalues $\{e_1, e_2, e_3\}$of Z_i set, therefore $Z_i' = V^T Z_i$.
(2) Delaunay triangulation. We calculate the normals of each triangle that form the segmented object $\{Z_i'\}$. An object contains a set of normals $N = \{n_1, n_2, \ldots, n_j\}$ where $n_j = [P_x^j, P_y^j, P_z^j]^T$.
(3) Finally, we use the directions of the normals to construct the histograms $H(\phi, \theta)$ of each object.

$$\text{Where } \phi = \tan^{-1}\left(\frac{P_x^j}{P_y^j}\right) \text{ and } \theta = \tan^{-1}\left(\frac{P_z^j}{\sqrt{P_x^2 + P_y^2}}\right)$$

3.2. Matching

Unknown objects are modelled by histograms H_i and match the histograms in the objects dataset H_b. Following the matching approach from [25, 26] we use

χ^2, histogram intersection, Haussler distance, euclidian distance, and earth mover's distance.

4. Results

In this section, we present the results of applying our method to detect vehicles on point clouds of urban environments.

4.1. Plane extraction

Planes are projected into lines as described in Section 1.1. We used the modified Hough transform to search for large sets of points that model lines. Each point on edge had an associated parametric line, the intersection of the parametric line designated the existence and position of collinear points. The higher the number of collinear points, the higher the probability of finding the plane. The extracted facades corresponded to parametric positions with several collinear points higher than 600 points in this stage. This parameter prevents side views of cars from being confused with facades. However, facades with collinear points less than 600 were not detected.

Table 1 shows the ground and perpendicular plane extraction equations where we used the modified Hough transform. In this table, we show the extraction of five perpendicular planes and the ground plane.

Follow the ground and the building facade removal; the remaining points contain unstructured objects and plane fragments.

4.2. Segmentation of unstructured objects

Our dataset vehicles have 1100 points on average; the sideways have 300 points on average. Therefore, the sideways of cars are not detected as facades. If multiple cars were aligned, the thresholds used to detect the facades would prevent the segmentation of these as a plane. Figure 2(a) shows a 3D points cloud where were extracted ground plane and facades. The remaining points correspond to trees, cars, persons, and small plane segments.

Figure 2(b) shows the segmented objects using Mean Shift, points belonging to the same object are painted with the same colour and labelled automatically (we repeat the colours to improve visibility). We used a threshold of 150 points to discard small objects that usually correspond to noise or far away objects and poorly defined. The segmentation result passes through the filtering stage and then to modelling and indexing.

4.3. Filtering

Once the unstructured objects such as pedestrians, trees, walls, lampposts, telephone booths, and cars are

Table 1. Extraction of structured objects.

Planes	*A*	*B*	*C*	*D*	# Points
Ground	0.0038	−0.0091	−1.0000	−177.05	27,278
Plane 1	0.2713	0.9626	0.0137	−611.30	13,957
Plane 2	−0.9122	0.4101	0.0097	404.75	5365
Plane 3	0.2867	0.9578	−0.0117	525.46	8793
Plane 4	−0.0010	0.0003	0.0000	2.3153	982
Plane 5	−0.9132	0.4074	0.0291	−673.52	2347
Unsegmented					21,939
Whole					80,661

Note: The point cloud has a total of 80,661 points, from which we extract 58,722 points belonging to the ground and facades. The plane equation is given by $AX + BY + CZ + D = 0$.

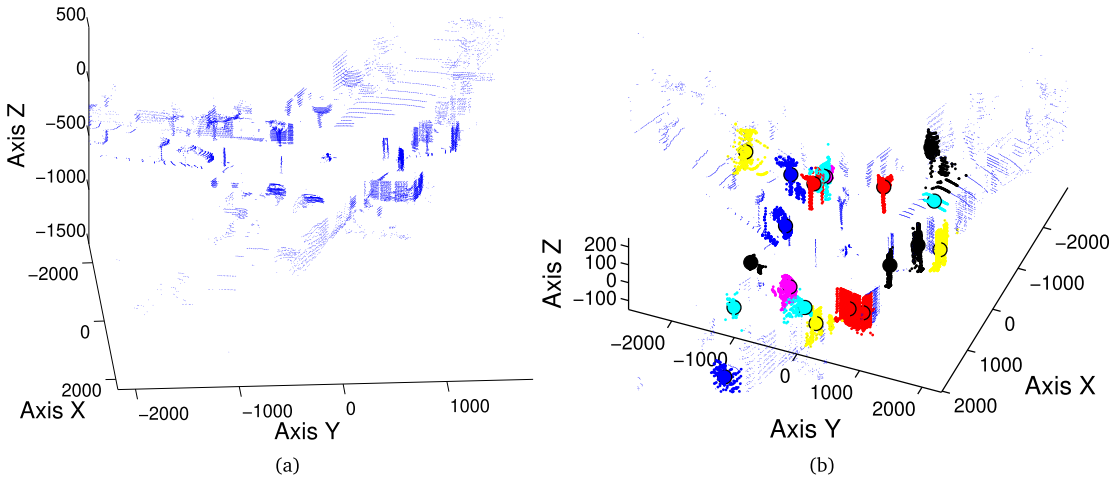


Figure 2. (a) Typical point cloud without structure objects. Due to the threshold values appear some points did not extract and belong to planes. (b) Mean Shift Segmentation-the points with the same colour belong to the same object. We repeat colours to improve visibility. We do not consider objects with a size less than 150 points.

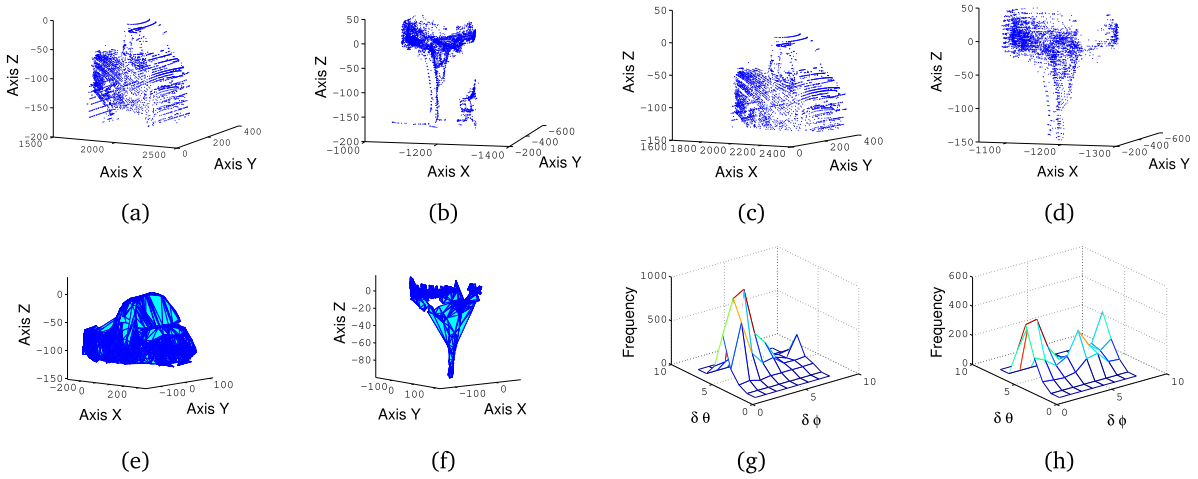


Figure 3. Filtering and Modeling. (a) Moreover, (b) show the points corresponding to a car and a tree, respectively, segmented via Mean Shift. (c) Furthermore, (d) show the same objects filtered via RPCA. (e) Moreover, (f) show the meshes obtained with Delaunay triangulation. (g) Moreover, (h) show the histograms generated with the normals of each triangle.

Table 2. Confusion matrix.

	χ^2 Statistics		Prediction		Histogram Intersection		Prediction
	86.50%	24.05%	Cars (P′)		87.69%	26.90%	Cars (P′)
	13.50%	75.95%	Other (N′)		12.31%	73.10%	Other (N′)
R.V.	Cars (P)	Other(N)		R.V.	Cars (P)	Other(N)	
	Haussler Distance		Prediction		Euclidean Distance		Prediction
	84.86%	28.48%	Cars (P′)		82.94%	21.52%	Cars (P′)
	15.14%	71.52%	Other (N′)		17.06%	78.48%	Other (N′)
R.V.	Cars (P)	Other(N)		R.V.	Cars (P)	Other(N)	
	Earth Mover Distance		Prediction				
	75.82%	26.90%	Cars (P′)				
	24.18%	73.10%	Other (N′)				
R.V.	Cars (P)	Other(N)					

Note: We used a car model as our reference and compared against the remaining objects on the test set. R.V is a real value.

segmented, they are passed through a filter stage to reduce noise. The 3D points acquired by the LiDAR Velodyne 64E are noisy. Several factors introduce the noise in 3D points; some are LiDAR-object distance, incidence angle, object-colour, and object-material [27].

Using ROC analysis, we computed the starting distance r of neighbours and the minimal neighbourhood points. The initial distance r is 80 cm, and the minimal number of points permit in a neighbourhood is 14. The filtering method allows a smooth surface and eliminates points that do not correspond to the object. Figure 3(a,b) show the objects before filtering and Figure 3(c,f) after filtering. We partition our dataset into training with 4500 objects and test with 3000 objects and optimize the Mean Shift thresholds,

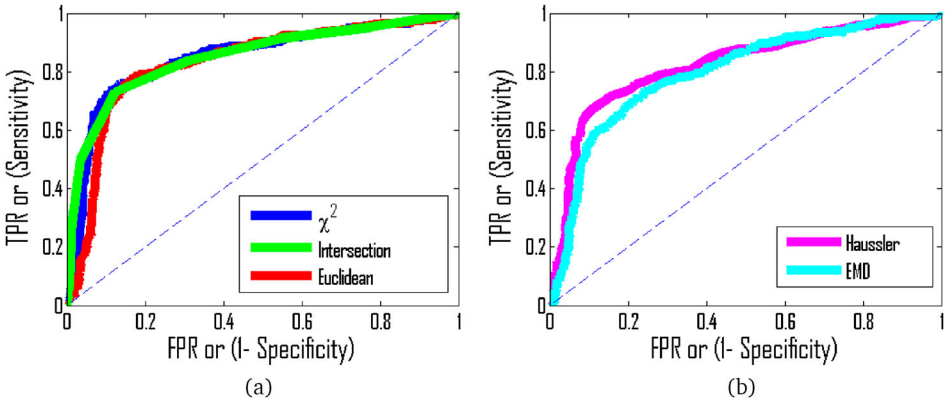


Figure 4. Performance Comparison. (a) In this plot, we show the best three methods: χ^2, Histogram Intersection, and Euclidean Distance. The area under the curve of the best method is 0.85 belonging to Histogram Intersection. (b) In this plot, we show the remaining methods: EMD and Haussler Distance.

the filter, the Delaunay parameters, and the histogram modelling via ROC analysis.

4.4. Modeling

After object filtering, we modelled the objects using histograms using the directions of the normal vector. Using Delaunay triangulation, we extracted the normal vector of each triangle to build the histograms, and we optimized the radius of the kernel filter of the Mean Shift method. Figure 3(c,d) show the filtered point clouds of a car and a three, respectively. Figure 3(e,e) show the same objects after triangulation. After triangulation, we can define the unique characteristics of each object based on its shape. Finally, Figure 3(g,h) show the histograms of the car and the tree.

We used four types of cars to improve detection: sedan, compact (size between 4 m and 4.7 m), SUV, and hatchbacks. By using different types of cars, we were able to optimize the algorithm and improve detection.

4.5. Matching

In this work, our interest is to detect cars. We partition our dataset into 4500 objects for training and 3000 objects for tests. Our objects included: cars, trees, lampposts, pedestrians, walls, or ground segments. Table 2 shows the confusion matrices after applying χ^2, histogram intersection, Haussler distance, Euclidean distance, and Earth Mover's distance on the test set. Figure 4(a,b) show the ROC curves for the five methods. The best method was Histogram Intersection, with an area under the curve (AUC) of 0.8501. We separated the ROC curves on two graphs to improve visualization. Table 3 shows the AUC for each of the five methods.

5. Discussion

To evaluate our proposal, firstly, we compare the final results modifying some of the steps that we consider most crucial, and we determined the best method for each step present in the flowchart of the Figure 1(b). As a second way to evaluate, we compare our results with other techniques proposed in the literature. The method proposed in this paper, differs from the ones in the literature as follows:

The flowchart of the Figure 1(b) shows three principal processes: Segmentation, 3D processing, and indexing. The segmentation process is out of the scope of this paper.. Table 3 shows the 3D processing and indexing process evaluation. Comparing the second and third columns, we can see that the final results are improved when we filter the 3D points of the segmented object. We observe in Table 3 that the matching step's best metrics are χ^2, and the intersection.

In the context of 3D city reconstruction [1, 28], we joined 3D data acquired in different positions to better define the object in the urban environment. However, there are several methods for urban object recognition that use the database KITTI. According to the size, truncation, and occlusion classes of objects, authors classify the object in the database KITTI into three difficulty classes: easy, moderate, and hard. Figure 5 shows the car examples of our database classify our dataset in easy, moderate, and hard. Table 4 shows the number of objects in our database for each class.

We compare our method with other state-of-the-art approaches on the car class of the KITTI validation set for 3D detection when the authors use the easy car class, and Average Precision (AP) Intersection-over-Union (IoU) 0.7 threshold. Table 5 shows that the car scan quality affects our method to a lesser degree when compared with state-of-the-art methods.

Table 3. The second and third columns show the performance similarity metrics for each of the five metrics corresponding to the matching step.

Metric	AUC, procedure shows in Figure 1(b)	AUC, without objects filter step
χ^2	0.8460	0.8130
Intersection	0.8501	0.8060
Haussler distance	0.8221	0.8114
Euclidean distance	0.8287	0.8045
Earth Mover's distance	0.7950	0.7798

Note: The second column shows the results using the proposed procedure in Figure 1(b). The third column shows the results without the 3D processing (Objects Filtering) step shown in Figure 1(b).

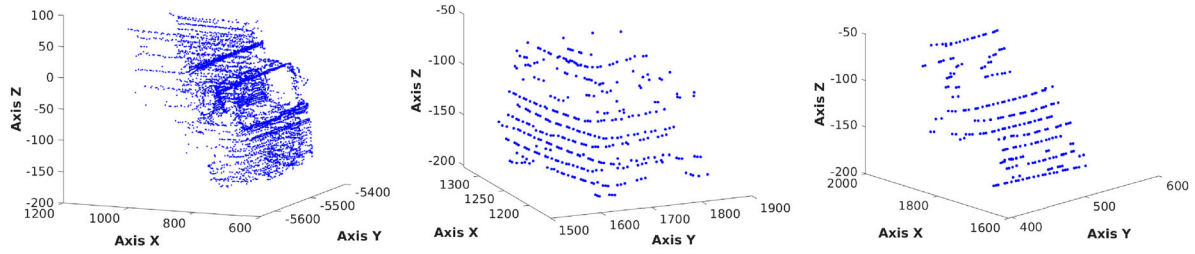
Table 4. Number of objects in our database.

Cars			Trees			Others objects			
Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Total
Trainnig									
197	590	392	26	68	17	476	1022	1712	4500
Test									
130	392	260	18	46	12	318	682	1142	3000

6. Conclusion and future works

In this work, we developed a new method for car detection on LiDAR point clouds. Our method has three parts: segmentation, filtering, and indexing. Segmentation rules out points belonging to facades and ground, keeping the remaining objects such as cars, trees, pedestrians, lampposts. Filtering improves the quality of the segmented objects by removing outliers. Indexing models the objects based on histograms of normal directions. We used a dataset acquired with our acquisition platform mounted on the top of a car and driving around the city for training and test. We partition our dataset in training and test, obtaining a detection rate of 85.01% on the test set using Histogram Intersection.

The future works in this research can be extended to the construction industry. For this application, we can be capture 3D point cloud data of construction sites,



(a) Easy (b) Moderate (c) Hard

Figure 5. Examples of the cars in our database classify into three difficulty classes: (a) easy, (b) moderate, and (c) hard.

Table 5. Comparison of our method with other state-of-the-art approaches on the car class of KITTI validation set for 3D detection.

		Automobile detection		
Method	Sensor	Easy	Moderat	Hard
Multi-View 3D [29]	LiDAR-Camera	71.29	62.68	56.56
PointFusion [30]	LiDAR-Camera	77.92	63.00	53.27
Frustum PointNets [31]	LiDAR-Camera	83.76	70.92	63.65
Deep Continuos Fusion [32]	LiDAR-Camera	86.32	73.25	67.81
AVOD Feature Pyramid [33]	LiDAR-Camera	84.41	74.44	68.65
IPOD [34]	LiDAR-Camera	84.10	76.40	75.30
Multi-Task Multi-Sensor Fusion [35]	LiDAR-Camera	87.90	77.86	75.57
VoxelNet [36]	LiDAR	81.97	65.46	62.85
SECOND [37]	LiDAR	87.43	76.48	69.10
BirdNet [38]	LiDAR	40.99	27.26	25.32
BirdNet+ [39]	LiDAR	70.14	51.85	50.03
SAPRNET [40]	LiDAR	84.92	75.64	67.70
Voxel-FPN [41]	LiDAR	85.64	76.70	69.44
Ours	LiDAR	77.68	71.73	72.58

works, or equipment to enable better decision making in construction project management. In future work, we will create semantically 3D models from point cloud data; object recognition must be labelled on point cloud data into object classes, e.g. wall, roof, floor, column, beam, Etc.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

To the CONACYT, for supporting project number 669. We also want to thank Instituto Politecnico Nacional , for supporting project SIP-20210280.

ORCID

Ramirez-Pedraza Alfonso http://orcid.org/0000-0003-0366-6249
González-Barbosa José-Joel http://orcid.org/0000-0002-6720-8282
Hurtado-Ramos Juan-Bautista http://orcid.org/0000-0003-2663-2463

References

[1] García-Moreno AI, Gonzalez-Barbosa JJ, Ornelas-Rodríguez FJ, et al. Automatic 3D city reconstruction platform using a LiDAR and DGPS. Oct. 2012. San Luis Potosi, Mexico. In: Advances in Artificial Intelligent, MICAI. 2013

[2] Zhou QY, Neumann U. Fast and Extensible Building Modeling from Airborne LiDAR Data. Nov 5. Irvine, CA, USA. In: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS). 2008.

[3] Sun S, Salvaggio C. Aerial 3D building detection and modeling from airborne LiDAR point clouds. IEEE J Sel Top Appl Earth Obs Remote Sens. 2013;6(3):1440–1449.

[4] Sohn G, Jwa Y, Kim HB. Automatic powerline scene classification and reconstruction using airborne LiDAR data. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci. 2012;I-3:167–172. DOI:10.5194/isprsannals-I-3-167-2012

[5] Himmelsbach M, Luettel T, Wuensche HJ. Real-time object classification in 3D point clouds using point feature histograms. October 15. St. Louis, MO, USA. In: Intelligent Robots and Systems (IROS). 2009.

[6] Wang C, Xu D, Zhu Y, et al. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. Jan 15. Long Beach, CA. In: The IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[7] Chen J, Cho YK, Kira Z. Multi-view incremental segmentation of 3D point clouds for mobile robots. CoRR. 2019. abs1902.06768.

[8] Danielczuk M, Matl M, Gupta S, et al. Segmenting unknown 3D objects from real depth images using mask R-CNN trained on synthetic point clouds. CoRR. 2018. abs/1809.05825. Available from: http://arxiv.org/abs/1809.05825.

[9] Jørgensen TB, Jensen SHN, Aanæs H, et al. An adaptive robotic system for doing pick and place operations with deformable objects. J Intell Robot Syst. 2019 Apr;94(1):81–100. ISSN 1573-0409. DOI:10.1007/s10846-018-0958-6

[10] Ji SQ, Huang MB, Huang HP. Robot intelligent grasp of unknown objects based on multi-sensor information. Sensors. 2019;19(7):1595. DOI:10.3390/s19071595

[11] Durović P, Grbić R, Cupec R. Visual servoing for low-cost scara robots using an rgb-d camera as the only sensor. Automatika. 2017;58(4):495–505. DOI:10.1080/00051144.2018.1461771

[12] Goron L, Tamas L, Reti I, et al. 3D laser scanning system and 3D segmentation of urban scenes. In: International Conference on Automation, Quality and Testing, Robotics (AQTR); Vol. 1. IEEE; 2010. p. 1–5.

[13] Guizzo Erico, Google. 2011. How Google's Self-Driving Car Works. IEEE Spectrum.
[14] Garcia F, Ponz A, Martín D, et al. Laser scanner and computer vision fusion for pedestrian detection in road environment. Rev Iberoam Autom In. 2015;12(2):218–229. DOI:10.1016/j.riai.2015.02.006
[15] Steder B, Grisetti G, Burgard W. Robust place recognition for 3D range data based on point features. In: International Conference on Robotics and Automation (ICRA). IEEE; 2010. p. 1400–1405.
[16] Sipiran I, Bustos B. Harris 3d: a robust extension of the harris operator for interest point detection on 3D meshes. Vis Comput. 2011;27(11):963–976.
[17] Xiao W, Zaforemska A, Smigaj M, et al. Mean shift segmentation assessment for individual forest tree delineation from airborne LiDAR data. Remote Sens. 2019;11(11):1263. DOI:10.3390/rs11111263
[18] Hernández-García DE, González-Barbosa JJ, Hurtado-Ramos JB, et al. 3D city models: mapping approach using LiDAR technology. March 1 San Andres Cholula, Mexico. In International Conference on Electronics, Communications and Computers (CONIELECOMP). 2011.
[19] BIPM. Evaluation of measurement dataguide to the expression of uncertainty in measurement. JCGM 100; 2008.
[20] Atanacio-Jiménez G, González-Barbosa JJ, Hurtado-Ramos JB, et al. Lidar velodyne hdl-64e calibration using pattern planes. Int J Adv Robot Syst. 2011;8(5):70–82.
[21] Leal E, Leal N, Sánchez G. Estimación de normales y reducción de datos atípicos en nubes de puntos tridimensionales. Inf tecnol. 2014;25(2):39–46.
[22] Hubert M, Rousseeuw PJ, Vanden Branden K. Robpca: a new approach to robust principal component analysis. Technometrics. 2005;47(1):64–79.
[23] Larrey-Ruiz J, Morales-Sánchez J, Larrey-Ruiz L. Efficient combined ssim- and landmark-driven image registration in a variational framework. Rev int métodos numér cálc diseño ing. 2019;35(1):1263.
[24] Zhang M, Fu R, Guo Y, et al. Moving object classification using 3D point cloud in urban traffic environment. J Adv Transp. 2020;12(2):1–12. DOI:10.1155/2020/1583129
[25] Gonzalez-Barbosa JJ, Lacroix S. Rover localization in natural environments by indexing panoramic images. August 07.Washington, DC, USA . In IEEE International Conference on Robotics and Automation. 2002.
[26] Luber M. People tracking under social constraints [dissertation]. Universitätsbibliothek Freiburg; 2013.
[27] Bolkas D, Martinez A. Effect of target color and scanning geometry on terrestrial LiDAR point-cloud noise and plane fitting. J Appl Geod. 2018 Jan;12(1):109–127. ISSN 1862-9016. DOI:10.1515/jag-2017-0034
[28] García-Moreno A, González-Barbosa J. Reconstrucción virtual tridimensional de entornos urbanos complejos. Rev Iberoam Autom In. 2020;17(1):22–33. ISSN 1697-7920. DOI:10.4995/riai.2019.11203
[29] Chen X, Ma H, Wan J, et al. Multi-view 3D object detection network for autonomous driving. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 6526–6534. DOI:10.1109/CVPR.2017.691
[30] Xu D, Anguelov D, Jain A. Pointfusion: deep sensor fusion for 3D bounding box estimation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); Los Alamitos, CA, USA. IEEE Computer Society; 2018 Jun. p. 244–253. DOI:10.1109/CVPR.2018.00033
[31] Qi CR, Liu W, Wu C, et al. Frustum pointnets for 3D object detection from rgb-d data. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. p. 918–927. DOI:10.1109/CVPR.2018.00102
[32] Liang M, Yang B, Wang S, et al. Deep continuous fusion for multi-sensor 3D object detection. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer vision – ECCV 2018. Cham: Springer International Publishing; 2018. p. 663–678. ISBN 978-3-030-01270-0.
[33] Ku J, Mozifian M, Lee J, et al. Joint 3D proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2018. p. 1–8. DOI:10.1109/IROS.2018.8594049
[34] Yang Z, Sun Y, Liu S, et al. Ipod: intensive point-based object detector for point cloud. Preprint; 2018. arXiv:1812.05276.
[35] Liang M, Yang B, Chen Y, et al. Multi-task multi-sensor fusion for 3D object detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 7337–7345. DOI:10.1109/CVPR.2019.00752
[36] Zhou Y, Tuzel O. Voxelnet: end-to-end learning for point cloud based 3D object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. p. 4490–4499. DOI:10.1109/CVPR.2018.00472
[37] Yan Y, Mao Y, Li B. Second: sparsely embedded convolutional detection. Sensors. 2018;18(10). ISSN 1424-8220. DOI:10.3390/s18103337
[38] Beltrán J, Guindel C, Moreno FM, et al. Birdnet: a 3D object detection framework from LiDAR information. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC); 2018. p. 3517–3523. DOI:10.1109/ITSC.2018.8569311
[39] Barrera A, Guindel C, Beltrán J, et al. Birdnet+: end-to-end 3D object detection in LiDAR bird's eye view. Preprint; 2020. arXiv:2003.04188.
[40] Ye Y, Chen H, Zhang C, et al. Sarpnet: shape attention regional proposal network for LiDAR-based 3D object detection. Neurocomputing. 2020;379:53–63. ISSN 0925-2312. DOI:10.1016/j.neucom.2019.09.086
[41] Kuang H, Wang B, An J, et al. Voxel-fpn: multi-scale voxel feature aggregation for 3D object detection from LiDAR point clouds. Sensors. 2020;20(3). ISSN 1424-8220. DOI:10.3390/s20030704