MSc in Photonics

Universitat Politècnica de Catalunya (UPC)
Universitat Autònoma de Barcelona (UAB)
Universitat de Barcelona (UB)
Institut de Ciències Fotòniques (ICFO)

http://www.photonicsbcn.eu

PHOTONICS**BCN**

Master in Photonics

MASTER THESIS WORK

DETECTION OF ROAD MARKINGS FROM LIDAR DATA

Xavier Pérez Moré

Supervised by Dr. Carlos Fernández López (KIT) and Dr./Prof. Santiago Royo Royo (UPC)

Presented on date 31th August 2022

Registered at

Detection of road markings from LIDAR data

Xavier Pérez Moré

MRT (Institut für Mess- und Regelungstechnik), Karlsruher Institut für Technologie, Engler-Bunte-Ring 21, Gebäude 40.32, D-76131 Karlsruhe, Germany

E-mail: `xavier.perez.more@estudiantat.upc.edu`

Abstract.

It is not rare to find LIDAR technology (Light Detection and Ranging) in the sensor setup of an autonomous car, since it can offer many features and compensate lackings of cameras. Since LIDAR sensors measure reflectivity, they can be useful in road markings detection methods. Many different techniques have been explored to take advantage of their higher reflectivity in relation to other road elements. An algorithm for the detection of road markings from LIDAR pointclouds is proposed and tested on the *KITTI* raw dataset. An augmentation of the pointcloud size using the data from the navigation sensor and a filter to detect road points are applied to ensure a good detection thanks to a thresholding performed based on the reflectivity values. It is proven that indeed detection from LIDAR measurements is possible regardless of the external light conditions.

For future work, a deep learning approach is encouraged to classify the detected road markings.

Keywords: LIDAR, road markings detection, autonomous driving

1. Introduction

Lane information is key in the understanding of traffic codes. Road marks give high priority information to the driver as the actual lane delimitation, the lanes the car can occupy, crosswalks, stop signs and many others. Therefore, a high precision road markings detection algorithm is fundamental for driverless cars to navigate safely.

Cars not only rely on camera systems but also other kinds of sensors like microwave radar or ultrasonic sensors. Each have strenghts and weaknesses, so sensor systems are usually diversified. In the last years, the emergence of laser based sensors has brought new options and improvements to autonomous cars perception systems [1].

These are better known as LIDAR, which stands for "Light Detection and Ranging". Its working principle consists of measuring properties of scattered light in order to find information of surrounding objects such as range, or velocity [2]. LIDAR sensors emit rapid laser pulses in high spatial-temporal resolution to the environment and a sensor measures the time it takes for them to scatter back . Therefore LIDAR generates a three-dimensional pointcloud of the surrounding objects as it can calculate at high precision their range from the sensor.

LIDAR presents many advantages over other sensing technologies, such as high accuracy, less interferences or the ability to provide information to interpret complex environments thanks to its 3D nature and a greater range [3]. But among these strengths there is one that stands out for our purpose. Road marks have a significantly higher reflectivity than the road surface. Therefore, LIDAR technology is ideal for road marks recognition since it measures this quantity directly.

In this article, a road markings detection algorithm aimed to exploit this property is presented.

1.1. Why LIDAR

Besides convenience for LIDAR detection based methods, highly reflective lane markings are optimal for both human drivers and machine vision. Lane paint is usually colored white, yellow or a color that presents a great contrast to the road surface, however reflectivity needs to be enhanced so the readability and perception of the marks is still possible in low-light and night-time conditions [4].

Road marks are designed to be retroreflective [5]. Such surfaces reflect light directly back to the light source itself. Generally retroreflective paint has a very dense distribution of glass beads with a 10-100 μm radius [6]. The beads embedded in the paintings must be round and transparent so a portion of light is reflected back to the illumination source [6].

1.2. State of the art

There also exist other detection techniques based on different technology sensors. The most popular alternative would be detection from RGB or greyscale cameras and it can be achieved by clustering pixels by color characteristics or brightness respectively [7] [8]. However, in these techniques external illumination now does play a part, so while under good lighting conditions results can be as good as LIDAR based methods' results, shadows can be a threat for example.

Furthermore, the data obtained from cameras is bidimensional, so getting a bird eye view projection as we do with the LIDAR data can be more complicated. An homography transformation ought to be done and still marks can appear somehow distorted.

However these methods present the advantage that the sensor setup is simpler and cheaper. Self driving cars need to have cameras either way, and a LIDAR sensor can be expensive.

Regarding LIDAR, the majority of methods are based on adaptive thresholding of the intensity values [9]. Since the advantage that LIDAR presents is that it does not depend on external light conditions, it is the first approach that comes to mind.

2. Methods

The structure of the method is the following: First of all, the LIDAR pointcloud for each scan is augmentated aiming at higher datapoints density (2.2). Secondly a height filter is applied in order to extract road surface points from outlying elements as vegetation, sidewalks or cars (2.3). In the next step, a bird eye view projection of the 3D LIDAR points is made onto the z plane. The core of the method consists in filtering the road points by their reflectivity so points with values above a threshold are labeled as road marks' points(2.4). Finally a clustering algorithm (DBSCAN) is applied to remove highly reflective noise datapoints in order to detect the actual road marks (2.5). A sketch of the process is shown in figure 1.

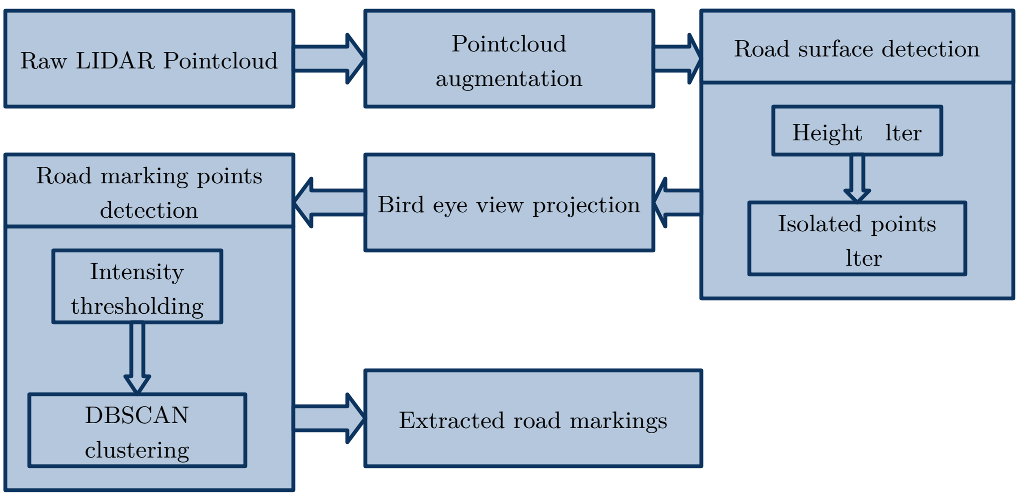

Figure 1. Diagram showing the algorithm structure.

A *Python* code has been built from scratch that carries out the algorithm described. To acces the raw *KITTI* data, some tools from a development kit from Lee Clement and his group from the University of Toronto have been used (`https://github.com/utiasSTARS/pykitti.git`).

2.1. Dataset

The KITTI raw dataset [10] has been used for this project. It includes data from multiple color and greyscale stereo cameras, a *Velodyne* ® 3D laser scanner and a high-precision GPS/IMU inertial navigation system taken within the city of Karlsruhe, Germany.

Thanks to the LIDAR sensor, a Velodyne ® HDL-64E rotating 3D laser scanner three-dimensional points with its reflectivity are obtained. The navigation system consists of an OXTS ® RT3003 inertial and GPS sensor. The setup is illustrated in figure 2.

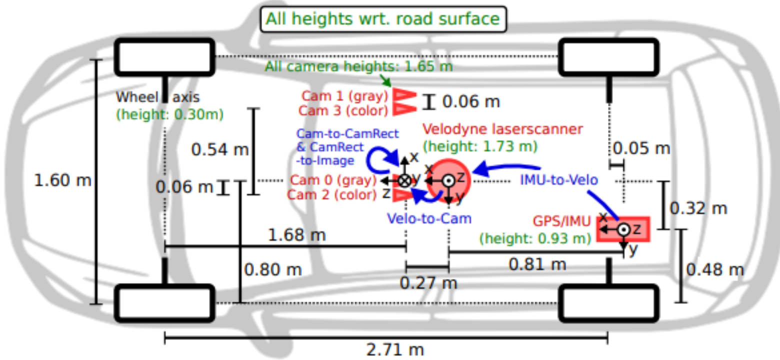

Figure 2. Illustration of the sensor setup of the car. Reproduced from [10]

We have opted for the raw dataset rather than the Semantic KITTI dataset [11]. Even though the latter presents the advantage of labeling road and non road points, the data has been taken mostly in residential areas of Karlsruhe, where there are barely any road marks. It is not that the detection tool would not work, but a more lane markings abundant dataset is more adequate for this endeavor. Then, good results in the segmentation would be useful for the training of a neural network, for example.

The tested scans are from the “City” and “Road” categories for the same reason, as “Residential” and “Campus” categories sequences contain very few road marks.

2.2. Pointcloud augmentation

In order to get better results, the LIDAR pointcloud has been augmented. The best technique to achieve so consists in adding to each 360 degrees LIDAR scan the data points from the previous and posterior scans in time, properly translated and rotated according to the movement of the vehicle.

Each scan contains approximately 120,000 points. To each frame, the data points from the 8 scans taken previously and after that are incorporated, adding up to an approximate total of 2,040,000 points per frame. All points' x,y,z coordinates from the previous and posterior dataframes are multiplied by a transformation matrix, as the vehicle movement from the added scan to the scan where points are added has to be taken into consideration, otherwise the added points will be shifted.

The transformation can be described as following:

(i) The translation matrix is calculated from the latitude, longitude and altitude values from the OXTS sensor. Their increment from the origin point is calculated and translated to the Velodyne® coordinates system.

(ii) The rotation matrix is generated from the roll, pitch and yaw angles from the OXTS ® sensor.

(iii) The pose formalism has ben used, that stands for position and orientation, often adopted in robotics. In three dimensions, the transformation matrix from the pose of a point B to a point A is as following:

$$
{}^{A}T_{B} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}t_{B} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{A}t_{Bx} \\ r_{21} & r_{22} & r_{23} & {}^{A}t_{By} \\ r_{31} & r_{32} & r_{33} & {}^{A}t_{Bz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

Being the inverse transformation:

$$
{}^{A}T_{B}{}^{-1} = \begin{bmatrix} {}^{A}R_{B}{}^{T} & -{}^{A}t_{B} \\ 0 & 1 \end{bmatrix} \tag{2}
$$

And the pose of the point transformated can be obtained as:

$$
{}^{A}\tilde{p} = {}^{A}T_{B}{}^{B}\tilde{p} \tag{3}
$$

$$
\begin{bmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}t_{B} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \\ 1 \end{bmatrix} \tag{4}
$$

As a result, a much denser pointcloud is generated so the lane marks are much more defined and its shapes are much better reproduced.

2.3. Height filter

Several issues appear if the intensity thresholding is performed directly after the last step. The motivation of this approach lies in the fact that reflectivity values from the road markings points will be higher than the road ones. However, no assurance is granted that elements apart from the road cannot be as reflective or more than the road markings. Vegetation, signs, or other vehicles can originate highly reflective points for example. Moreover, laser rays can interact with suspended elements as dust specks resulting in misreadings of the distance, for instance. For many more reasons, points not belonging to lane marks can fall above the threshold. These events can mess with the effectiveness of the adaptive threshold criteria set. In order to reduce these errors some filters are implemented.

To tackle the problem of excluding non-road elements from the intensity thresholding, there is a wide variety of options. Many researchers opt for a curb detection algorithm based on the change of slope that LIDAR scan rings experiment due to change of altitude of the edge of the road [9], or directly deep learning approaches. However, better results have been found with an altitude filter. We exploit the tridimensionality of the LIDAR data, and an altitude threshold is set on the z coordinate measurements, so the points below are classified as road points. The threshold is not adaptive, since the position of the sensor within the car and the altitude of the car are known, we can determine the relative position between the LIDAR and the ground at all times.

Besides that, in case still some isolated highly reflective points remain apart from the lane markings, an isolated points filter is also applied. Those points whose distance to their adjacent points overcome a certain threshold are overlooked.

2.4. Intensity thresholding

Once the pointcloud is filtered so ideally only points belonging to the road are left, the intensity thresholding takes place. Even though lane marks reflectivity values should stay pretty consistent regardless of external conditions, as stated throughout the report, there might be some variations due to other factors, such as the actual state of the markings. Paint can be more or less degraded for example.

The optimal procedure consists in setting an adaptive threshold. The mean and standard deviation from all points' intensity have been calculated. All points with an intensity above the mean plus the standard deviation value multiplied by 2.5 are labeled as road mark points.

2.5. DBSCAN

Finally, a clustering method has been applied in order to remove noise and detect the lane marks as units. The Density-Based Spatial Clustering of Applications with Noise algorithm [12], referred from now on as DBSCAN is implemented through a *Python* library.

The DBSCAN algorithm is an unsupervised learning method that operates as following.

- It creates an epsilon radius circle centered on each point.
- A point is classified as a "Core" point if its circle contains a chosen minimum number of points, as a "Border" point if the number does not reach the minimum number, and as "Noise" if there are not any data points within the circle.

It is the best for our purpose since it is very effective at handling outliers and separating clusters with a high density of points from low density ones. Furthermore it presents the advantage that the number of clusters does not need to be set in advance.

The algorithm is very sensitive to epsilon and the minimum number of points for cluster. In our case we work with a very small epsilon value, since every point from a road mark should be very close to another point of the same road mark.

3. Results and discussion

The pointcloud augmentation method proposed proves to be a huge improvement since every scan generates a very sparse pointcloud. However, adding more than 10 scans approximately ends up defeating the purpose, as the marks can become blurry and lose definition. This probably happens because for larger car movements the error in the transformation coordinates given by the navigation sensor is higher. In order to avoid that, only 8 posterior and 8 previous scans are added to the pointcloud.

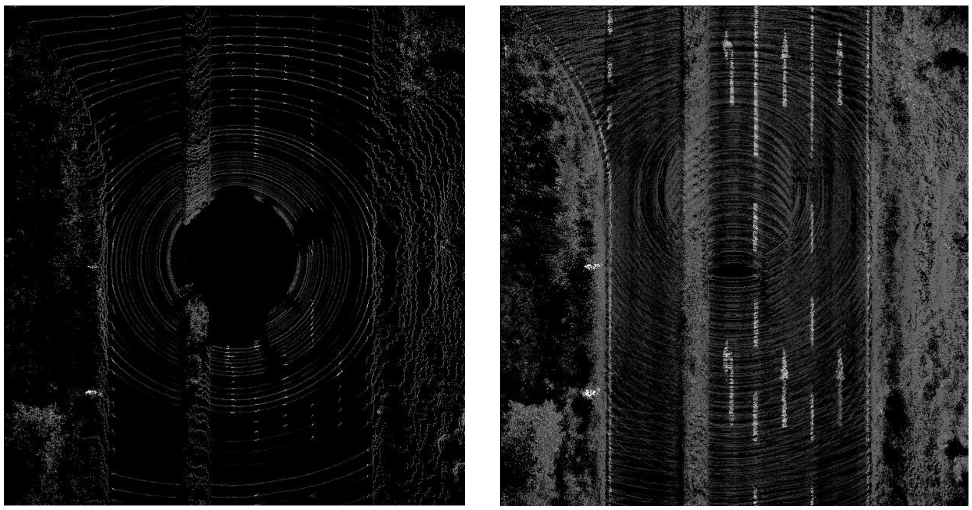

Figure 3. Comparison of the bird eye view of a LIDAR pointcloud before and after the data augmentation is carried on. The projection is represented with a colormap that pictures reflectivity for each point. Points with a 1.0 reflectivity value are white and those that are not reflective are black. Values in between are colored in a greyscale

The results are satisfying since road markings feature are very defined and distinguishable from the road.

On the other hand, the algorithm that separates road from environment is built on the assumption that there is a change of altitude in the surface between road pavement and the non-drivable area. However, despite being that the usual scenario, it is not necessarily like that. It also can happen that the slope variation between road and non-road is minimal and not sensitive to the threshold. Either way, this part is the one that can present more problems from the whole algorithm. Nonetheless, most of these problematics are corrected with the noise removing clustering algorithm.

Some other issues arise regarding the height filter. For example, it is less efective for the points that are further from the sensor. This is probably due to a combination of errors from the LIDAR distance measurements for longer range, and again an effect of higher errors in the translation data from the IMU/GPS sensor for bigger displacements. One solution would be to limit the range of the points considered. Even though not significantly, this compromises the adaptability of the method.

Thanks to thee DBSCAN algorithm, which offers a robust clusterization, noise is eliminated very effectively for an epsilon value of 6 cm and a minimum number of points per cluster of 20.

Finally, the results from the algorithm are presented in figures 4 and 5. For each frame depicted, the LIDAR pointcloud bird eye view projection is represented as a 2D greyscale heatmap where white accounts for high reflectivity and black for low reflectivity. The detected road markings have been highlighted over the pointcloud map in yellow. For better understanding of the scene, an RGB image taken at the same time by one of the car's cameras is shown next to the markings representation.

Some hypothesis have been verified such as the detection performance not being affected by the external illumination conditions. In figure 4 it can be seen how shadows and abrupt illumination variations do not impede the lane marks detection.

It can also be observed that there are yellow dots that do not correspond to road markings outside of the road, like in figure 5, evidencing the shortcomings of the height filter explained earlier. On an ending note, in both frames represented the road markings are effectively detected, and in figure 5 the pursued detail and accuracy in position of the road marks detected can be appreciated.

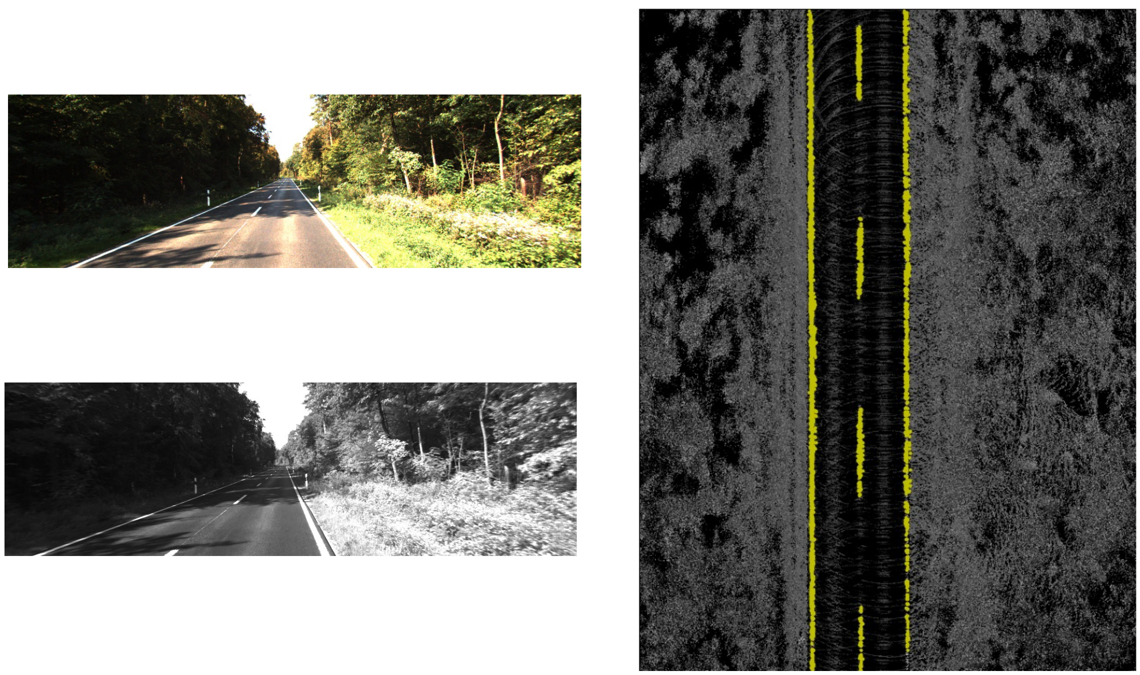

Figure 4. Road markings detected highlighted over greyscale representation of the pointcloud. In the left there are images of the same scene taken by an RGB and a greyscale camera respectively.

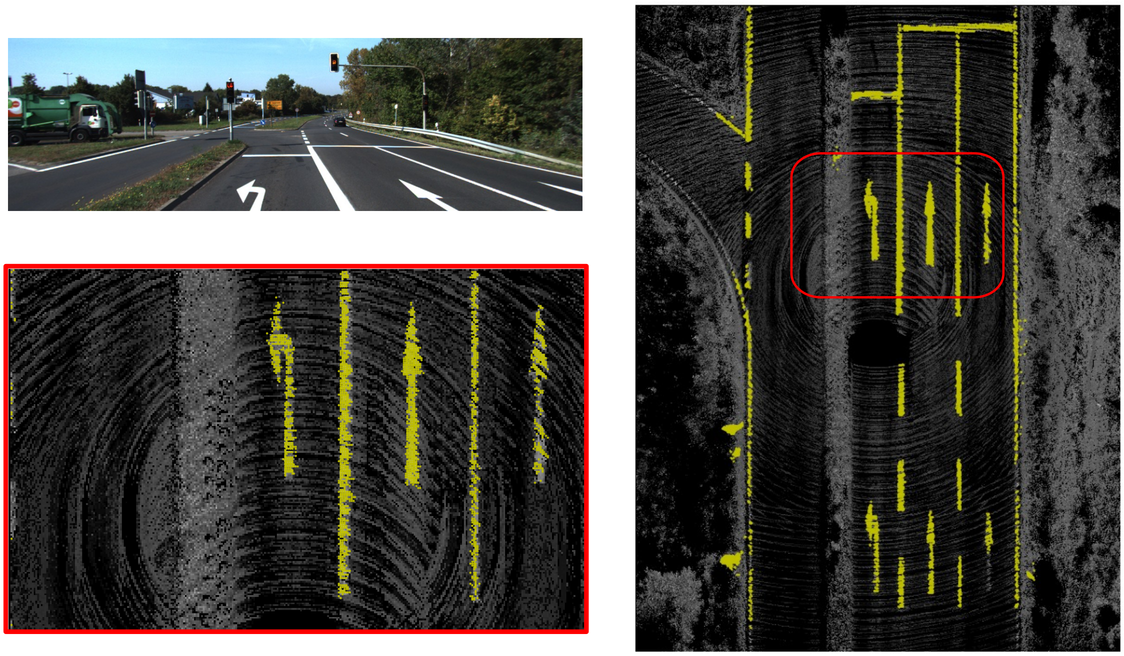

Figure 5. Road markings detected highlighted over greyscale representation of the pointcloud. In the left there is an image of the same scene taken by an RGB camera and a close up look to some markings

4. Conclusions

LIDAR pointclouds have been demonstrated to offer a good visualization and to enable an effective detection of road markings, based on their reflectivity values. Moreover, the performance of the algorithm is not affected by external illumination conditions.

However, the algorithm presents some limitations. The height filter shortcomings result in either a reduction of the effective range of detection or lead to errors in the detection. The first scenario is not a very good option since for higher car velocities the detection range might fall short. Better results would be achieved with a more precise road delimitation technique or a semantic dataset where road points are already labeled [11].

Future work could be headed in the direction of using the detection tool to get a ground truth for road markings points segmentation for the *KITTI* dataset. If this ground truth is good enough, it can be used to train a neural network to detect road markings. Furthermore, given the high definition of the road marks detected thanks to the pointcloud augmentation, it would be interesting to explore the option of using a neural network trained to classify different road markings in arrows, lane lines, crosswalks and all possible kinds on the result of the segmentation.

References

[1] Jeff Hecht. Lidar for self-driving cars. *Optics and Photonics News*, 29(1):26–33, 2018.

[2] Patrick Chazette, Julien Totems, Laurent Hespel, and Jean-Stéphane Bailly. Principle and physics of the lidar measurement. In *Optical Remote Sensing of Land Surface*, pages 201–247. Elsevier, 2016.

[3] Ciyun Lin, Yingzhi Guo, Wenjun Li, Hui Liu, and Dayong Wu. An automatic lane marking detection method with low-density roadside lidar data. *IEEE Sensors Journal*, 21(8):10029–10038, 2021.

[4] Tan Li and Deng Zhidong. A new 3d lidar-based lane markings recognition approach. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2197–2202, 2013.

[5] Erzhuo Che, Michael J Olsen, Christopher E Parrish, and Jaehoon Jung. Pavement marking retroreflectivity estimation and evaluation using mobile lidar data. *Photogrammetric Engineering & Remote Sensing*, 85(8):573–583, 2019.

[6] Darko Babić, Tomasz E Burghardt, and Dario Babić. Application and characteristics of waterborne road marking paint. *International Journal for Traffic and Transport Engineering*, 5(2):150–169, 2015.

[7] Zhao Li, Zi-xing Cai, Jin Xie, and Xiao-ping Ren. Road markings extraction based on threshold segmentation. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1924–1928. IEEE, 2012.

[8] Trung-Thien Tran, Chan-Su Bae, Young-Nam Kim, Hyo-Moon Cho, and Sang-Bock Cho. An adaptive method for lane marking detection based on hsi color model. In *International Conference on Intelligent Computing*, pages 304–311. Springer, 2010.

[9] Alberto Hata and Denis Wolf. Road marking detection using lidar reflective intensity data and its application to vehicle localization. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 584–589. IEEE, 2014.

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[12] Jian Hou, Huijun Gao, and Xuelong Li. Dsets-dbscan: A parameter-free clustering algorithm. *IEEE Transactions on Image Processing*, 25(7):3182–3193, 2016.