Semi-automatic road lane marking detection based on point-cloud data for mapping

Zhen Kang 1∗, Qiao Zhang 2
1 School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan, Hubei, 430048, China
2 School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan, Hubei, 430048, China
∗Corresponding author’s e-mail: kang12229@whpu.edu.cn

Abstract. Traditional manual method for mapping is time consuming and has low precision. In order to increase the efficiency, this paper proposes a semi-automatic road lane marking detection based on point-cloud data. This approach requires simple interface interaction, and since point cloud data is memory intensive, we convert it to RGB images using the intensity and height of the 3D point cloud. Then, in the interface, the user needs to click a point around the detection line and generate image regions of interest. Finally, the image is filtered using a specific filter to obtain a convolved value for each point, and the position with the largest convolution value is the point at the position of the reticle. In this way, we get a point on the road marking, and multiple point connections generate the road markings on the map. Multiple experiments were performed on data from different routes obtained by different sensors, such as veledyne-64, veledyne-16, to verify the effectiveness of road lane marking detection based on point cloud data for mapping.

1. Introduction
Maps are very important in our lives. In an unfamiliar place, the navigation system will guide us on the right according to the map. In the automotive industry, maps are critical in autonomous driving and road intelligence systems. However, road markings are the most important signs on the map, including lane markings, zebra crossings and arrow markings. There are many studies on the use of various sensors for road marking detection [1-3]. Lee proposed a method of mixing LIDARS lane marking information with AVM cameras [2], which combines two different types of sensors in a uniform manner to achieve better performance, and is not only applicable to highways, but also Complex urban roads for slow speeds are quickly maneuvered in tight spaces. Hechri and Adellatif proposed a multi-tasking driver assistance system that includes two video-based driving tasks [3], lane detection and road sign recognition. Aly [4] proposed a real-time and robust method to detect lane markings in city streets using cameras. A hierarchical lane detection algorithm is proposed to deal with structured and unstructured roads [5]. Borkar describes a simple method for lane detection using Hough transforms and iterative matched filters [6-7], which creates a road aerial view by combining inverse perspective maps and applies random sample consensus to help eliminate roads. The outliers caused by noise and artifacts, as well as a Kalman filter, help smooth the output of the lane tracker. Guan Ziyun, Li Zongsheng proposed a method that uses a variety of data features of laser scanning to detect and extract road markings [8]. Ammu and Philomina discuss the lane detection and tracking algorithms of
the past decade [9]. Alberto Hata [10] proposed a road marking detection method based on the Otsu threshold method, which can segment the LIDAR point cloud into asphalt and road markings.

We propose a filter-based road lane marking detection method that solves point cloud data collected by laser scanning. The filter kernel used in this method is generated based on the characteristics of the road marker and is unique. The filter quickly determines the position of the possible reticle. The following are the steps of the method. In the first step, the user needs to select a point around the road marker, based on which we can obtain the image ROI for detection. The second is to project the point cloud data onto the 2D RGB image by meshing method according to its height and intensity. A pre-processing method such as image filtering, image thresholding, etc. is then applied to enhance the contrast of the image, wherein the filtering kernel is artificially generated based on the features. Finally, we build the kd tree to search for the nearest neighbor, which is the best point.

Several experiments were performed on different test routes and different laser scans (e.g., veledyne-64, veledyne-16) to verify the performance of lane marker detection based on point cloud data for mapping. The method is simple and fast, and is suitable for simple and flat road scenes, which can help people draw maps quickly. The method is simple in operation, simple in operation, and low in requirements on a computer.

2. Methodology

2.1. Projection

The point cloud data not only contains three-dimensional coordinates, but also include reflection intensity information, wherein the intensity information is related to the surface material of the target, the roughness, the direction of the incident angle, and the emission energy of the instrument and the laser wavelength. Therefore, it has a strong reference value in road marking detection. The point cloud has a large amount of data and is redundant. The more common processing method in the application is to project the 3D point cloud data onto the 2D image plane. The projection method in this paper is based on the plane where the ground in the point cloud is located, and projects the three-dimensional data onto the plane, that is a front view projection. It is prescribed that the front of the radar is in the positive direction of the X-axis, the left side is the positive direction of the Y-axis, and the vertical direction is the positive direction of the Z-axis. The algorithm first discards all point clouds whose height value exceeds a certain threshold. Because we detect the marks, edges and road edges on the road, we only need very low height data. The point cloud data whose ground height exceeds a certain threshold includes a lot of interference information, such as Power poles, trees, billboards, viaducts, etc., can cause erroneous detection results. Then, the obtained point cloud is divided into small rectangular grids, and all the points in the grid are projected to obtain a pixel on the image. This forms the projected image. The following is the specific implementation of the projection.

We use the height and intensity information of the pavement in the point cloud to detect road lane markings, instead of processing the point cloud directly. We project the 3D point-cloud to 2D geo-referenced RGB image according to the height and intensity information of the data points by gridding method [8]. First, we set an initial point and extract the features of point-clouds, such as max height, min height, max intensity, point number and so on. Then we assign the R, G and B values of the projection image as the count, the max intensity and the height of this grid respectively. We need to pay attention to the following points in the projected process. The x, y coordinate points in image need to be shaped. The value of the pixel in the image needs to be shaped and between 0-255. The starting point of the image coordinate is in the upper left corner, and there is an offset from the laser radar origin, the coordinates of the point cloud may be far away and the picture size is limited, it is necessary to limit the projection in the interest area of point cloud. The initial point cloud ROI is showing in Figure 1 and we get a projected image as shown in Figure 2.
2.2. Preprocessing

Image filtering can both eliminate noise mixed in the image and extract image features for image recognition. There are many kinds of filtering methods, including nonlinear filtering, median filtering, morphological filtering, bilateral filtering, custom filtering, etc. The text is customized to filter the image according to the characteristics of the feature to be extracted.

In order to generate the filter kernel, we combined the literature [10] to analyze the characteristics of the road markings in the point cloud data, such as the line width, the curb height, the marking intensity and so on. This paper we detect three kinds of line, include edge line, label line and double yellow line. In order to reduce the time cost during processing, the filter kernel is created at the beginning of the program. First, we donate the kernel size as KEL_SIZE and image pixel as p(i, j), based on a large number of statistical results we set KEL_SIZE=79.

The initial filter kernel for edge line and label line as:

\[
p(i, j) = \begin{cases} 
1, & \text{if } (i = \frac{KEL_SIZE}{2} - 1 \text{ or } i = \frac{KEL_SIZE}{2}) \\
0, & \text{else}
\end{cases}
\]  

(1)

Since the filtering kernel of the double yellow line is relatively complicated, it is cumbersome to express it by formula, and here is shown by pseudo code.

**Algorithm 1: Create Filter Kernel for Double Line**

```plaintext
for i = -3 to 13
    set row = \frac{kel_{size}}{2} + i
    for j = 0 to kel_{size}/2
        set col = \frac{kel_{size}}{4} + j
        if i = 4 or 5
            p(i, j) = -0.5
        else if i = -2, -3, 11 or 12
            p(i, j) = -1
        else if i = -1, 2, 3, 6, 7 or 10
            p(i, j) = 0
        else
            p(i, j) = 1
```


Considering the uncertainty of the point cloud direction, based on the initial filter kernel obtained above, it is rotated in units of one angle to obtain 360 filter kernels in different directions, so that straight lines in all directions can be detected.

2.3. Lane Detection
This section describes the basic steps of the lane detection. With the projection method of the previous part, we get a RGB image, and then some image processing methods will be used for the reticle detection. First, we split the RGB image to three single channel images. According to different kinds of line detection described above, we calculate the least value of height, and if the height difference of the current point is greater than preset threshold, which is set to an invalid point. Second, we calculate a mask image, if the point is invalid, the mask value set as zero, others are set 255. Thirdly, we process the mask image by filtering, binary processing, and operation. At last, we find nonzero points for following optimization. The algorithm is showing in figure 3.

![Detection algorithm](image)

Figure 3. Detection algorithm.

2.4. Optimization
The nearest-neighbor search is often used to the best match, pattern recognition and so on [10]. In the optimization stage, kd-tree method is chose. In lane detection, we acquire some candidates for optimization. In this section, we build kd-tree structure and use knn-search method to calculate the nearest point. At last, we transform this point to 3D point cloud space.

3. Experimental Results
The point cloud data shown in the figure below is obtained on the country road by the veledyne-64 line laser. Figure 4 shows the initial point cloud data. Figure 5 shows the detection results of a point. Figure 6 shows the results of the road boundary detection. The points in the image are the initial points given manually, and the lines are the edges detected by the algorithm based on the initial points. When the road is curved, in order to increase the accuracy, dense points are required, as shown in Figure 6 (a). Figure 7 shows the results of road edge line detection. Figure 8 shows the results of road markings detection.

![The initial point cloud](image)

Figure 4. The initial point cloud.
Figure 5. The detection results of a point.

(a) Curved road boundary detection.  (b) Straight road boundary detection.

(c) Near straight road boundary detection

Figure 6. The results of the road boundary detection.

Figure 7. The results of road edge line detection.

Figure 8. The results of road markings detection.

The algorithm in this paper is easy to understand, fast in processing, and can view the results in real time, which can reduce manual operations, improve the accuracy of map drawing, reduce human error in the fast drawing map application, and have certain value in engineering applications.
4. Conclusion
This paper proposes a semi-automatic filter-based road marking detection algorithm, which is applied to fast map rendering. This is an efficient, real-time and robust way to help employees reduce drawing time and drawing errors. The method first extracts the collected 3D point cloud data and combines them into effective point cloud blocks through the driving track, then projects the point cloud data to generate images, and uses a custom filter to detect the required features. The nearest neighbor search algorithm is used to find the most advantageous position needed, which are the points on the line that need to be detected. The method is simple and fast, and is suitable for simple and flat road scenes, which can help people draw maps quickly. The method is simple in operation, and low in requirements on a computer. However, when the point cloud is too sparse or the road is severely damaged, the algorithm will fail and the improved algorithm will continue to be studied later.

Acknowledgement
This work was supported in part by WHPU under Grant 2018Y17.

References
[1] Khan, B. S. , Hanafi, M. , & Mashohor, S. (2016). In: Automated road marking detection system for autonomous car. Research & Development.
[2] Lee, H. , Kim, S. , Park, S. , Jeong, Y. , Lee, H. , & Yi, K. (2017). AVM / LiDAR sensor based lane marking detection method for automated driving on complex urban roads. In: IEEE Intelligent Vehicles Symposium (IV).
[3] Ahmed Hechri, Abdellatif (2011). Lanes and Road Signs Recognition for Driver Assistance System. In: IJCSI International Journal of Computer Science Issues, 8(6): 402-408.
[4] Aly, M. . (2008). Real time detection of lane markers in urban streets. In: Intelligent Vehicles Symposium
[5] Cheng, H. Y. , Yu, C. C. , Tseng, C. C. , Fan, K. C. , Hwang, J. N. , & Jeng, B. S. (2008). Hierarchical lane detection for different types of roads. In: IEEE International Conference on Acoustics.
[6] Borkar, A. , Hayes, M. , Smith, M. T. , & Pankanti, S. (2009). A layered approach to robust lane detection at night. In: IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems.
[7] Borkar, A. , Hayes, M. , & Smith, M. T. (2010). Robust lane detection and tracking with ransac and Kalman filter. In: IEEE International Conference on Image Processing.
[8] Hariyan Guan, Jonathan Li, Yongtao Yu, Cheng Wang (2013). Rapid Update of Road Surface Databases Using Mobile LiDAR: Road-markings. In: 2013 Fifth International Conference on Geo-Information Technologies for Natural Disaster Management.
[9] Ammu M Kumar, Philomena Simon (2015). Review of Lane Detection and Tracking Algorithms in advanced driver Assistance System. International Journal of Computer Science & Information Technology (IJCST), 7(4): 65-78.
[10] Alberto Hata, Denis Wolf (2014). Road marking detection using LiDAR reflective intensity data and its application to vehicle localization. In: 17th International IEEE Conference on Intelligent Transportation Systems. pp. 584-589.