

THE IMPLEMENTATION OF SEMI-AUTOMATED ROAD SURFACE MARKINGS EXTRACTION SCHEMES UTILIZING MOBILE LASER SCANNED POINT CLOUDS FOR HD MAPS PRODUCTION

Y.-F. Chang ^{1,a,*}, K.-W. Chiang ^{1,b}, M.-L. Tsai ^{1,c}, P.-L. Lee ^{1,d}, J.-C. Zeng ², N. El-Sheimy ³, H. Darweesh ⁴

¹ NCKU, Department of Geomatics, No. 1, Daxue Rd., East Dist., Tainan City 701401, Taiwan – ^a a12000feng@gmail.com, ^b kwchiang@geomatics.ncku.edu.tw, ^c taurusbryant@geomatics.ncku.edu.tw, ^d pointnpl@geomatics.ncku.edu.tw

² XUNWEI TECHNOLOGIES, 3 F., No. 57, Zhousi St., Neihu Dist., Taipei City 11493, Taiwan – jessica31017nsgs@gmail.com

³ Dept. of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4, Canada – elsheimy@ucalgary.ca

⁴ Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan – hatem.darweesh@zatitech.com

KEY WORDS: Autonomous Driving, High-definition Maps, Mobile Laser Scanning, Otsu Threshold Filter, Semi-automatic Algorithm, Road Surface Markings.

ABSTRACT:

As research on autonomous driving deepens, High-definition Maps (HD Maps) have gradually become an auxiliary information for the new generation of autonomous driving technology. Compared to traditional electronic navigation maps, HD Maps have higher accuracy requirements and more information. Multi-road environment information and road elements are included. In the production of HD Maps, the on-board Mobile Laser Scanning (MLS) system has the ability to quickly collect environmental information, with high precision, thus making the system a widely used data collection method today. However, subsequent map building, digitization, and other mapping work still rely on manual operation, which is time-consuming and laborious. Therefore, this research is dedicated to developing a semi-automatic algorithm to generate HD Maps from the acquired point cloud data. This research focuses on the extraction of road surface markings, using the Cloth Simulation Filter (CSF) to obtain the road surface point cloud to improve the extraction efficiency. The road markings are extracted using the characteristic of high intensity values, and the commonly used Otsu threshold filter in image processing is used to extract point clouds with high reflectance intensity, eliminating the need for manual setting of point clouds. And based on geometric conditions, the objects are classified, such as arrow lines, pedestrian crossings, stop lines, and lane lines, which are convenient for further mapping HD Maps.

1. INTRODUCTION

1.1 General Instructions

According to the Monitoring Progress in Urban Road Safety: 2022 Update, released by the World Health Organization(WHO) in 2022, an estimated 1.3 million people die on the roads worldwide every year, with 20-50 million people suffering non-fatal injuries. The main factor in traffic accidents is human error, such as driver distraction, excessive speed, and poor directional control (Papantoniou et al., 2019). As a result, the technology related to autonomous vehicles has received widespread attention from both the industrial and academic sectors. With the development of smart cities and Intelligent Transportation Systems (ITS), the problem of autonomous driving has recently received much attention. In addition to increasing road safety, autonomous vehicles can also reduce the stress and cost of drivers, increase road capacity, reduce energy consumption and pollution, and improve fuel efficiency (Litman, 2022).

Many car systems have started to integrate Advanced Driver Assistance Systems (ADAS), which provide drivers with information about the vehicle's operation and changes in the external environment. The ADAS then assists the driver in assessing the surrounding situation and issuing early warnings of potential dangers, allowing the driver to respond promptly and take appropriate measures. With the rise of intelligent unmanned vehicles, the development of self-driving technology requires HD Maps as a base for spatial information to ensure that the self-driving vehicles operate on the correct path. (Bock, 2021).

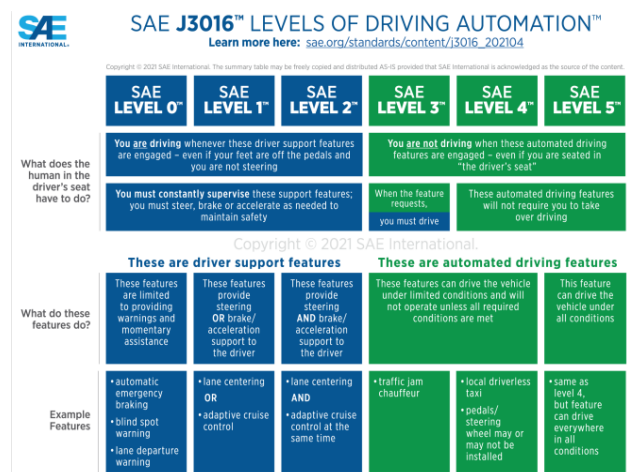


Figure 1. SAE levels of driving automation (Bock, 2021).

The Society of Automotive Engineers (SAE) has defined six levels for driving automation, as shown in Figure 1. Level 0 is non-autonomous driving, which means that the driver is required to drive the vehicle throughout the entire process. Levels 1 and 2 apply the functions of Advanced Driver Assistance Systems (ADAS) to reduce the burden of driving. However, the driver still needs to be fully focused during the driving process. For Level 4, the Automated Driving System (ADS) can independently perform all driving tasks, except in special circumstances, (NHTSA).

In recent years, with the development of self-driving cars, the sensors equipped on vehicles have become more diverse, such as cameras, LiDAR, and radar. However, these sensors also have their own limitations, such as being affected by lighting or cost. To improve navigation accuracy, HD Maps are used as additional auxiliary information and are less affected by external factors such as weather and other vehicles. HD Maps enable AD systems to surpass the view of traditional sensors, thus providing accurate and detailed information about the driving environment (Jeong et al., 2022). However, HD Maps are expensive to produce due to the use of expensive sensors and manual digitization in terms of data processing. Therefore, reducing the production cost of HD Maps is an important topic for the development of autonomous vehicles (Chiang et al., 2022). The automatic production of HD maps can significantly reduce labour and time costs. Semi-automatic procedures have been proposed for final evaluation and revision of the generated HD maps (Chiang et al., 2022), as shown in Figure 2.

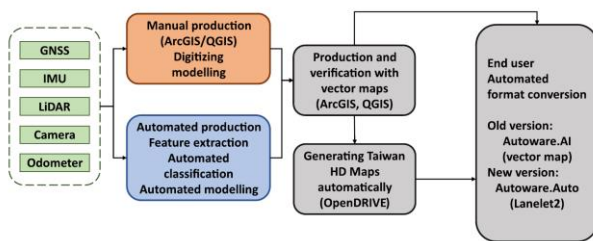


Figure 2. Semiautomated HD maps production procedure (Chiang et al., 2022).

Some studies have tried to use deep learning to extract road objects (Elhousni et al., 2020), but the extraction accuracy does not meet the accuracy requirements of HD Maps. In addition, in terms of data acquisition, MLS point clouds are currently the most widely used due to their high accuracy. Therefore, this study proposes a semi-automatic method to extract specific road elements from MLS point clouds to generate HD Maps based on geometric properties.

The accuracy of HD Maps is important for the development of autonomous vehicles. According to the Taiwan Information and Communication Standards Association, the accuracy requirements for HD Maps are 20 cm in the horizontal direction and 30 cm in 3D space. In this study, specific road elements such as arrow lines, pedestrian crossings, stop lines, and lane lines are extracted from MLS point clouds to generate HD Maps through a semi-automatic method. The results of this study will contribute to the development of more efficient and cost-effective methods for producing HD Maps, which will ultimately aid in the advancement of autonomous vehicle technology. The accuracy of the results is validated in three test fields and must meet the accuracy requirements of HD Maps. Therefore, the goal is to determine road surface regions, preserve surface point clouds, and identify road surface features. Some methods in the literature can achieve such a goal, with a common road surface extraction method based on curb structures using the height differences

between sidewalks and roads to find the lane and further obtain the road surface (Yang et al., 2013). However, not all fields contain curb structures. Additionally, there are also research methods that use the intensity differences between road and sidewalk point clouds to extract road edge information (Zeng, 2020); however, if the road boundary contains low-reflectivity materials such as soil, some errors may occur during the extraction process. Therefore, this study applies Cloth Simulation Filter (CSF) as an alternative road surface extraction method, which can fit the local terrain based on the ground, and is less affected by height differences and reflectivity.

Specifically, this paper aims to:

- Develop semi-automated algorithms for extracting specific road surface elements from point clouds.
- Implement a semi-automatic method on the MLS point cloud in the experimental field for the practical application of high-definition map feature object extraction.
- Evaluate the absolute accuracy of the modeling results using existing validated HD maps to ensure the quality of the extracted objects.

The structure of this paper is arranged as follows: In Section 2, the semi-automated extraction process and method are described, including the screening of road surface point clouds, the use of the Otsu threshold method (Otsu, 1979) for automatic threshold selection, and the identification of target objects based on their geometric characteristics. Section 3 provides an overview of the experimental setup. Section 4 presents the extraction results and precision. Finally, the paper concludes with Section 5, which summarizes the findings and highlights the key takeaways.

2. METHODOLOGY

To improve the efficiency of extracting road surface objects, this paper incorporates the Cloth Simulation Filter (CSF) and Otsu Threshold Filter for extracting road surface point clouds into the algorithm architecture. These methods can save a significant amount of time that would have been spent on manual editing. Section 2.1 will introduce the process of extracting road surface markings and lane lines. Section 2.2 will explain the data preprocessing while Section 2.3 will introduce the methods used in road surface marking extraction.

2.1 Extraction Flowchart of Road Surface Markings

The flowchart in Figure 3 shows the process of extracting road surface markings. The first step is to apply a Cloth Simulation Filter to the point cloud to obtain the road surface point cloud. Then, the input road surface point cloud is translated to the local coordinate system to reduce the impact of the number of digits on the algorithm. Next, the point cloud is downgraded to voxels to reduce the computational cost of subsequent algorithms. Otsu Threshold Filter is then used to automatically filter the threshold and divide the point cloud into two categories: asphalt road and road surface markings. Then, the obtained road surface markings point cloud is processed with a Statistical Outlier Removal (SOR) filter to remove scattered noise point clouds statistically. However, the three targets we need to extract

- pedestrian crossing, arrow line, and stop line - are all painted with white paint, so they have very high reflectivity. Therefore, Otsu Threshold Filter (Otsu, 1979) is used again to obtain road surface markings painted with white paint. Subsequently, the obtained white point cloud is separated into individual objects using the Euclidean distance clustering method, and the length and width of the object are calculated using the Oriented Bounding Box (OBB) method. The object is then classified based on the geometric characteristics of the length and width.

Figure 4 shows the flowchart of extracting lane lines. After the high-reflectivity point clouds are classified into pedestrian crossing, arrow lines, and stop lines, the point clouds belonging to these three categories are removed. The remaining high-reflectivity point clouds are the lane line point clouds. Through mathematical fitting, the obtained lane lines can be analyzed for accuracy with high-definition maps.

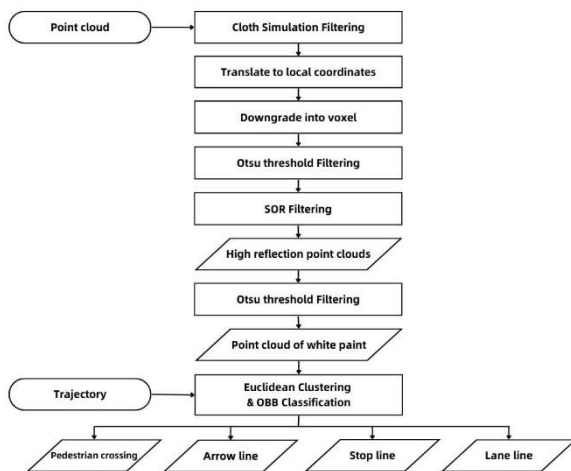


Figure 3. Flowchart of road surface marking extraction.

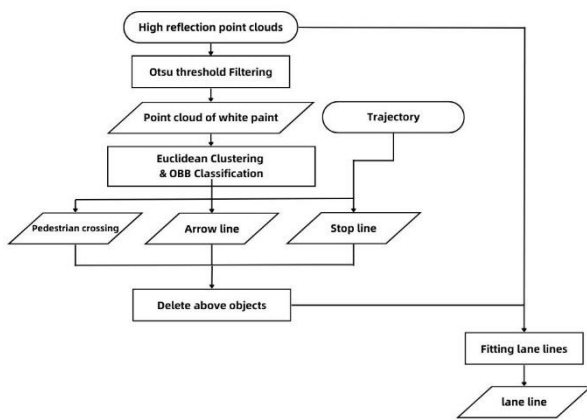


Figure 4. Flowchart of laneline extraction.

2.2 Point Cloud Preprocessing

Before performing object extraction, this research first applies Cloth Simulation Filter (CSF) to extract ground point clouds, as the original point clouds from MLS are very large. Then, the point clouds are downgraded to voxels and redundant point clouds are removed based on trajectory in order to increase the efficiency of the following algorithms. The following will

provide an explanation of the Cloth Simulation Filter (CSF) and the removal of redundant point clouds based on trajectory.

2.2.1 Cloth Simulation Filter (CSF): In this study, we use the Cloth Simulation Filter (CSF) method (Zhang et al., 2016) as a preprocessing step for efficiently extracting road surface objects from point cloud data. Traditional methods, which rely on elevation and slope as the filtering basis, are not suitable for complex and scene steep terrain areas. Therefore, we propose the use of CSF as an alternative method for data preprocessing.

The CSF method simulates the interaction between cloth nodes and corresponding lidar points, and uses the positions of the cloth nodes to generate an approximation of the surface. The method involves inverting the existing point clouds and simulating a natural falling point cloth to fall on the inverted point clouds. The surface of the cloth is then used as the datum for the ground point cloud, as shown by the red dotted line in Figure 5.

The parameters set by the CSF will affect the simulation results of the datum surface, including rigidity, cloth grid resolution, and classification threshold. The rigidity determines the rigidity of the cloth and the flatter of the extracted reference plane. In this research, we set the rigidity to "Flat" as the experimental area is a flat environment. The cloth grid resolution determines the density of cloth nodes used, and it is set to 0.1 meter. The classification threshold sets a threshold to represent the elevation difference between the point cloud and the obtained datum, it is set to 0.1 meter. And we use CloudCompareStereo v2.12 software to perform CSF.

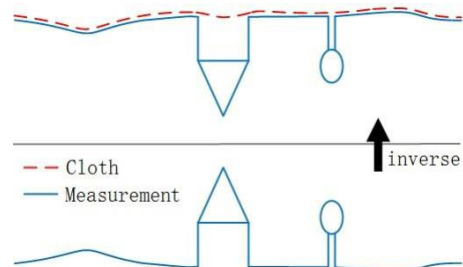


Figure 5. Cloth simulation filter.

2.2.2 Remove Redundant Point Cloud based on Trajectory:

In our study, we have implemented a filtering process for the lidar point clouds to remove some redundant data, such as those from surrounding grass or buildings. This is achieved by setting a threshold based on the scanning trajectory of the lidar and deleting any point clouds that are too far from this trajectory. The result of this point cloud filtering is shown in Figure 6, where Figure 6 (a) represents the point clouds before deletion and Figure 6 (b) represents the point clouds after deletion. The white points in the figure are the trajectory data we use. We set the threshold according to the road width of the experimental field to eliminate redundant point clouds. This process helps to reduce the amount of calculations required and improve overall performance.

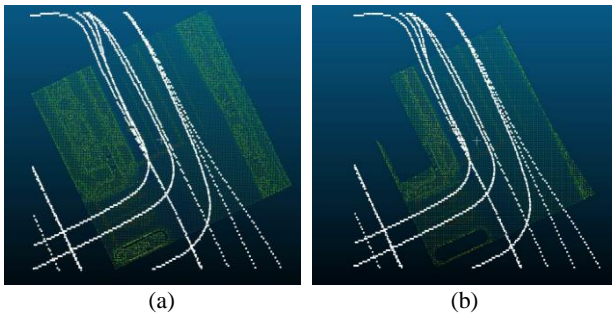


Figure 6. Flowchart of laneline extraction.

2.3 Road Surface Marking Extraction

After preprocessing, the point cloud data mostly consists of road surface point clouds, which can generally be divided into two categories: asphalt road surfaces and road markings. There is a significant difference in reflection intensity values between the two. Otsu Threshold Filter to automatically select the most suitable classification threshold, which replaces the manual selection of thresholds used in the past. After extracting the road markings, the Euclidean Distance Clustering method is used to group different objects. Before clustering, the discrete points will be removed using the Statistical Outlier Removal (SOR) filter. Finally, the Oriented Bounding Box (OBB) Algorithm is used to calculate specific lengths and widths for the road surface objects. However, stop lines are often connected to lane lines and are not easily grouped, so they are extracted with the assistance of trajectory. The detailed method will be explained in Section 2.3.5.

2.3.1 Otsu Threshold Filter: In extracting road surface markings, we filter by intensity values. This is because lines such as arrow lines, pedestrian crossing, stop lines, etc. that are painted with yellow or white paint have higher reflectivity values than asphalt roads. As shown in the reflection intensity value square diagram in Figure 7, an appropriate threshold can be selected to divide all point clouds into two categories. However, if we manually set an intensity threshold to distinguish road markings, it takes a lot of time to search for the threshold and it is difficult to set the optimal threshold at once. Therefore, in the extraction of road markings, we introduce the Otsu threshold method to set the threshold. The Otsu threshold method is a common binarization method used in image processing. The method uses an exhaustive search to find the optimal threshold that maximizes the variance between two classified classes by searching from the minimum value to the maximum value and selecting the threshold with the maximum variance between the foreground and background. However, this method is only effective when there is a significant difference in strength between the two categories, which is the case between road markings and asphalt roads. The formula for interclass variance is shown in Equation 1:

$$g = N_f \times N_b \times (I_f - I_b)^2, \quad (1)$$

where N_f = the number of points in the foreground
 N_b = the number of points in the background
 I_f = mean reflection intensity of foreground
 I_b = mean reflection intensity of background

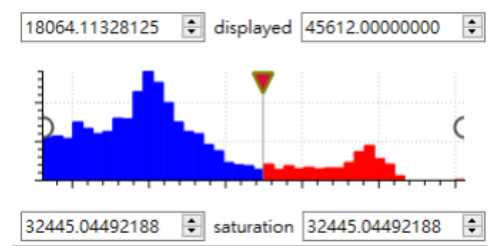


Figure 7. The histogram of point clouds reflection intensity value.

2.3.2 Statistical Outlier Removal (SOR) Filter: Road markings typically have higher intensity values than paved roads, and LiDAR intensity values are less affected by external factors such as weather and lighting (Gwon et al., 2017), making LiDAR point cloud data suitable for automated extraction. However, even after applying an intensity-based threshold, scattered outliers may still exist in the point cloud data. These outliers may be caused by various sources such as noise from the scanning process or errors in measurement and can negatively impact subsequent clustering-based point cloud classification.

To address this issue, this study proposes the use of Statistical Outlier Removal (SOR) filter as a processing step to remove scattered outliers before extracting road surface objects. The SOR filter uses statistical analysis to identify outliers by calculating the average distance of each point to its neighboring points, and then uses a threshold to identify points with distances far greater than their neighbors. These points are considered outliers and are removed from the dataset. The threshold can be set based on user preference or application.

The results of this study show that using SOR filter as a processing step effectively removes scattered outliers, improving the accuracy and reliability of subsequent clustering-based point cloud data classification.

2.3.3 Extract and Classify White Objects: In this study, we use the Euclidean clustering algorithm to further extract specific road markings such as white arrows, crossing lines, and stop lines. By using the Otsu threshold method again, we extract the white road markings with high reflection intensity. These specific markings have distinct geometric features, such as specific shapes and sizes, which we use to filter the corresponding road markings.

The Euclidean clustering algorithm groups all road point clouds into different clusters based on semantic and topological information, as shown in Figure 8 (Jiang, 2017). This process begins by marking all road point clouds as non-clustered points. Then, a random seed point is selected, and its neighboring point clouds are searched within a given radius (referred to as the Euclidean threshold). After the clustering search is completed, another seed point is selected from the remaining non-clustered points and the process is repeated until all point clouds are grouped into clusters.

This method allows us to effectively separate road markings into different clusters, separating the target point clouds we wish to extract into different groups.

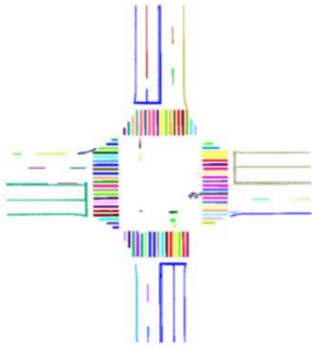


Figure 8. Illustration of clustered road markings (Jiang, 2017).

2.3.4 Oriented Bounding Box (OBB) algorithm: The road surface markings after point cloud clustering can be classified into various categories according to the geometric information of each cluster. This study refers to road marking design standards, adjusts the parameters of the algorithm, and distinguishes the types of road markings by the geometric shapes of various road markings. To obtain the geometry of each cluster, we generate a minimum bounding box to wrap all point clouds in each cluster to determine the width and length of the cluster. The reason this method works is because bounding boxes are usually regular and more efficient for calculation, representing the object with a bounding box instead of using the real shape of the object. Among the types of bounding boxes, Axis-Aligned Bounding Box (AABB) and Oriented Bounding Box (OBB) are commonly used bounding volume types (Gottschalk et al., 2000). The AABB is easy to construct because it is aligned with the three axes in the current coordinate system. However, the AABB does not rotate with the wrapped object. It results in the bounding box not tightly surrounding the object. In contrast, OBB generates a rectangular bounding box that follows the principal components of the object, which means that any plane of the bounding box does not have to be parallel to any of the three axes in the current coordinate system, as shown in Figure 9. It can describe the target object more accurately. The principal components of each bounding box, that is, the principal components of each cluster, can be obtained by the Principal Component Analysis (PCA) algorithm. PCA is known as a data analysis method that transforms raw data into a set of linearly independent representations in each dimension by linear transformation, and reduces the dimensionality of the dataset by identifying the most salient directions (Dimitrov et al., 2006).

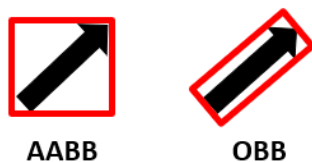


Figure 9. Axis-aligned bounding box (AABB) and oriented bounding box (OBB).

After OBB calculates the length and width of the object, we will classify it according to the road marking design standard. Table 1 sorts out the geometric characteristics of the target object.

Type	Note
Pedestrian crossing	Width = 0.4 m 2 m < Length < 8 m
Arrow lines	0.9 m < Width < 2.3 m 4 m < Length < 5 m
Stop lines	0.3 m < width < 0.4 m

Table 1. The geometric definition of road markings

2.3.5 Extraction of Stop Lines: However, even after extracting pedestrian crossing lines and arrow lines using the Euclidean clustering algorithm and OBB algorithm, there are still issues with identifying stop lines. This is because stop lines are often connected to lane lines. Therefore, by applying a distance-based Euclidean clustering method, these different types of road markings are assigned to the same cluster. In this paper, we use a geometric condition to differentiate between stop lines and lane lines. Under normal circumstances, the direction of the test vehicle platform is perpendicular to the stop line. Therefore, we use track data to divide the road markings into blocks to search for stop lines. As shown in Figure 10, if the density of points in the red block is higher than a given density threshold D_T , the points within the block are likely to be stop lines. The condition for identifying stop lines is as following Equation 2:

$$\begin{cases} \text{if density} = \frac{P_B}{W_B \times L_B} > D_T, \text{ block} \in \text{stop line points} \\ \text{otherwise, other road marking points} \end{cases}, \quad (2)$$

where P_B = the number of points in the block
 W_B = the width of the block
 L_B = the length of the block

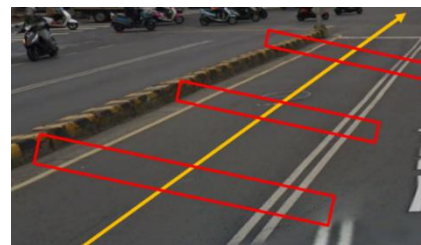


Figure 10. Trajectory-based search for stop lines.

As shown in Figure 10, the yellow arrow represents the vehicle trajectory, and the red box represents the search box of the stop line.

However, due to the poor accuracy of stop line extraction, this study additionally uses manually corrected trajectories, which can make up for the weakness of the algorithm in this study in accurately extracting stop lines.

3. EXPERIMENT SETUPS

3.1 Experimental Environmental Description

The Taiwan CAR LAB in Shalun, Tainan is an ideal location for research and development in autonomous driving technology. As the first closed-field test site in Taiwan, it offers a controlled environment for the evaluation and refinement of algorithms related to self-driving cars. The site features 13 simulated traffic scenarios, including challenging situations such as railway level crossings, detours, and tunnels, which provide valuable data for researchers. One of the key advantages of the

CAR LAB is the clearly marked lines and the absence of other vehicle interference, which allows for more accurate and reliable testing. The simulated traffic scenarios also mimic real-world conditions, providing a more realistic testing environment.

Overall, CAR LAB Taiwan provides a unique opportunity for researchers to develop and test autonomous driving technology, mapping technology in a controlled and realistic environment. Scenes from this site are shown in Figure 11.



Figure 11. Taiwan CAR LAB (CARTURE , 2019).

4. RESULTS AND DISCUSSION

4.1 Extraction Result

4.1.1 Otsu Threshold: After applying the Otsu threshold method, all point clouds are classified into two categories: low reflection intensity and high reflection intensity. As shown in Figure 12, the point clouds with high reflection intensity values have been successfully filtered out and isolated. These point clouds, which correspond to the road surface markings, can then be further analyzed using clustering algorithms and geometric characteristics to distinguish between different types of markings. To enhance the accuracy of the classification, additional image processing techniques such as morphological operations may also be applied to refine the point clouds. Additionally, statistical methods such as principal component analysis (PCA) can also be used to extract and analyze the geometric features of the point clouds, which can aid in the classification process.

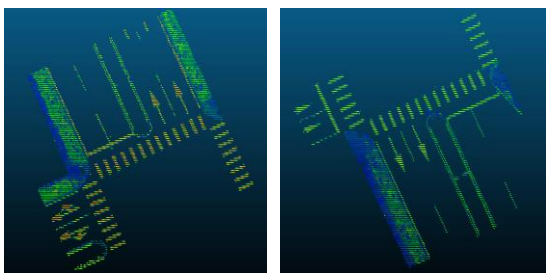


Figure 12. Extraction results of arrow lines, pedestrian crossing, and stop lines.

4.1.2 Road Marking Extraction: The Figure 13 shows the results of road markings extracted by our algorithm. The white points in the figure represent pedestrian crossing, red points represent arrow lines, and yellow points represent stop lines. The accuracy of the extracted road markings is high.

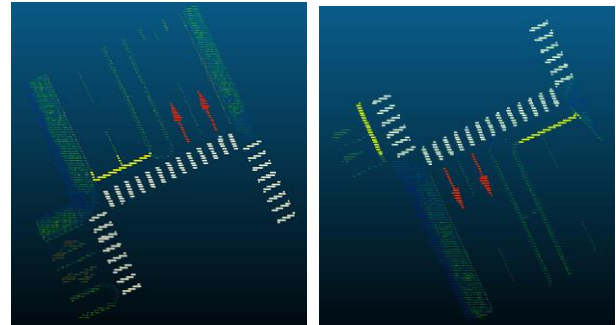


Figure 13. Extraction results of arrow lines, pedestrian crossing, and stop lines.

4.1.3 Lane Lines Extraction: In the clustering process using the Euclidean clustering method, stop lines and lane lines are often grouped together. The results in Figure 14 show that the stop lines are extracted by using the track assistance method, and the remaining point cloud is then clustered again, and the lane lines are fitted by mathematical interpolation.

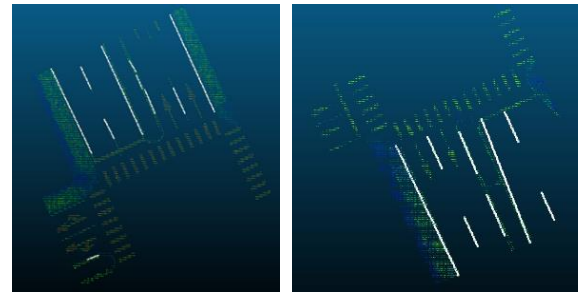


Figure 14. Extraction results of lane lines.

4.2 Accuracy Assessment

In this study, validation is conducted to evaluate the accuracy and performance of the proposed modeling methods. To ensure the reliability of the results, the reference data is taken from vector maps digitally recorded by a surveying company. The lane lines are depicted as polylines in shapefile format, which is commonly used in Geographical Information Systems (GIS) software. However, this format is not intuitive and it is difficult to assess the results automatically. Therefore, the shapefile is first converted into a CAD file. Then, the vector maps are divided into one point per centimeter, and the TWD 97 coordinates of each point are generated. The results are then compared to the reference data through statistical analysis, including the Root Mean Square Error (RMSE), Standard Deviation (STD), Mean Value, and Maximum Value. The equations for RMSE and STD are as follows Equation 3 and Equation 4:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (p_i - P_i)^2}{n}}, \quad (3)$$

$$STD = \sqrt{\frac{\sum_{i=1}^n (\hat{\varepsilon} - \varepsilon_i)^2}{n-1}}, \quad (4)$$

where p_i ($i=1,2,\dots,n$) indicates the coordinate of the modeled results, P_i ($i=1,2,\dots,n$) denotes the true coordinate of the corresponding modeled points from the reference data, ε_i ($i=1,2,\dots,n$) represents the true error of p_i and P_i and $\hat{\varepsilon}$ represents the mean value of the true errors. Through this

statistical analysis, the validation results demonstrate the absolute precision in the scale of the real world.

For the accuracy validation of arrow lines and pedestrian crossings, we used vector files provided by the surveying company as ground truth. We first visually inspected the extracted point clouds to check for any missing points, and then overlaid our algorithm's extracted point clouds the ground truth to calculate the proportion of points that fall within the ground truth. We used the ArcMap 10.5 (GIS software) for the calculations, and in the three experimental areas, the proportion of arrow lines and pedestrian crossings extracted by our algorithm that fall within the ground truth is over 95%, as shown in Table 2.

	Field 1		Field 2	
	Overlaid on HD map / Extracted point clouds	Rate (%)	Overlaid on HD map / Extracted point clouds	Rate (%)
Pedestrian Crossing	899/912	98.6%	814/834	97.6%
Arrow Lines	105/109	96.3%	120/126	95.2%

Table 2. Statistical analysis of pedestrian crossing and arrow lines

	Field 1		Field 2	
	2D	3D	2D	3D
Root Mean Square Error (RMSE) (m)	0.072	0.085	0.200	0.203
Average Error (m)	0.053	0.067	0.158	0.163
Max Error (m)	0.228	0.241	0.538	0.539
Standard Deviation (STD) (m)	0.049	0.053	0.123	0.120

Table 3. Accuracy analysis of extraction results of lane lines

The results of the lane line extraction, as shown in Table 3, demonstrate that the accuracy is in compliance with the requirements of 20 cm in 2D and 30 cm in 3D in all three experimental fields. Notably, the 3D accuracy in field one and two reaches less than 10 cm.

On the other hand, the modelling effect of the stop line is poor. This is because our algorithm will misjudge some lane lines as stop lines when extracting stop lines. Therefore, it is easy to make mistakes when fitting the end points of the stop line. In order to make up for the problem of excessive stop line extraction error, we use the manually corrected trajectories. We also analyzed the extraction accuracy of the stop line in two of the experimental fields, as shown in Table 4. The accuracy of this method meets the requirements for building HD Maps.

In conclusion, the proposed algorithm architecture in this study can extract elements required for HD Maps with high accuracy that meets the precision demands of HD Maps, and it saves a significant amount of manual digitization cost in a semi-automated way.

	Field 1		Field 2	
	2D	3D	2D	3D
Root Mean Square Error (RMSE) (m)	0.091	0.111	0.083	0.118
Average Error (m)	0.078	0.172	0.063	0.103
Max Error (m)	0.168	0.172	0.244	0.245
Standard Deviation (STD) (m)	0.049	0.105	0.055	0.058

Table 4. Accuracy analysis of extraction results of stop lines by using manually corrected trajectories.

5. CONCLUSION

The study of HD Maps production is currently in a flourishing stage. The high cost of sensors and the large amount of manual digitization make producing HD Maps very expensive. And reducing the proportion of manual involvement is the focus of research in this field. Due to efficient data collection and precise 3D geographic space measurement, commercial MLS has been widely used in the collection of data for HD Maps generation.

To reduce the cost of manual digitization, this thesis presents a semi-automated method for extracting and modeling specific road elements, including pedestrian crossings, arrow lines, parking lines, and lane lines, to generate HD Maps. Since the target elements are all part of the road surface's high-reflectance point clouds, this study uses the Cloth Simulation Filter for ground point filtering and the Otsu threshold method for automatically selecting the most suitable road marking threshold.

Additionally, this research tested the proposed method in three commercial MLS point cloud experiment areas using HD Maps vector files from a surveying company as ground truth. The results showed that the extraction accuracy of pedestrian crossings and arrow lines can reach above 95%. The modeling results for lane lines and stop lines meet the requirements of HD Maps, with a horizontal accuracy of less than 20 cm and a three-dimensional spatial accuracy of less than 30 cm. This study aggregates the semi-automation and aims to generate HD Maps that can be directly applied to future autonomous driving systems.

ACKNOWLEDGEMENTS

The author sincerely appreciates the technical assistance from Ms. Jih-Cing Zeng and the professional advices from the supervisor, Dr. Kai-Wei Chiang. The MOI of Executive Yuan, ROC (Taiwan) is acknowledged for providing financial support for this study. This study cannot be completed without their assistance. Besides, the author also wants to acknowledge Dr. Hatem Darweesh from Nagoya University. The author cannot finish the research without Dr. Hatem's professional assistance.

REFERENCES

- Bock, T., 2021. Unsettled Issues on HD Mapping Technology for Autonomous Driving and ADAS.
- Chiang, K.-W., Pai, H.-Y., Zeng, J.-C., Tsai, M.-L., El-Sheimy, N., 2022. Automated Modeling of Road Networks for High-Definition Maps in OpenDRIVE Format Using Mobile Mapping

Measurements. *Geomatics* 2, 221–235.
<https://doi.org/10.3390/geomatics2020013>

Chiang, K.W., Zeng, J.C., Tsai, M.L., Darweesh, H., Chen, P.X., Wang, C.K., 2022. Bending the Curve of HD Maps Production for Autonomous Vehicle Applications in Taiwan. *IEEE J Sel Top Appl Earth Obs Remote Sens* 15, 8346–8359.
<https://doi.org/10.1109/JSTARS.2022.3204306>

Dimitrov, D., Knauer, C., Kriegel, K., Rote, G., 2006. On the Bounding Boxes Obtained by Principal Component Analysis.

Elhousni, M., Lyu, Y., Zhang, Z., Huang, X., 2020. Automatic Building and Labeling of HD Maps with Deep Learning.

Gottschalk, S., Manocha, D., Lin, M.C., Brooks, F.P., 2000. Collision Queries using Oriented Bounding Boxes.

Gwon, G.P., Hur, W.S., Kim, S.W., Seo, S.W., 2017. Generation of a Precise and Efficient Lane-Level Road Map for Intelligent Vehicle Systems. *IEEE Trans Veh Technol* 66, 4517–4533.
<https://doi.org/10.1109/TVT.2016.2535210>

Jeong, J., Yoon, J.Y., Lee, H., Darweesh, H., Sung, W., 2022. Tutorial on High-Definition Map Generation for Automated Driving in Urban Environments. *Sensors* 22.
<https://doi.org/10.3390/s22187056>

Jiang, H., 2017. Semi-automated Generation of Road Transition Lines Using Mobile Laser Scanning Data.

Otsu, N., 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076

Papantoniou, P., Yannis, G., Christofa, E., 2019. Which factors lead to driving errors? A structural equation model analysis through a driving simulator experiment. *IATSS Research* 43, 44–50. <https://doi.org/10.1016/j.iatssr.2018.09.003>

Yang, B., Fang, L., Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 79, 80–93.
<https://doi.org/10.1016/j.isprsjprs.2013.01.016>

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens (Basel)* 8.
<https://doi.org/10.3390/rs8060501>