AUTOMATED MODELLING OF ROAD FOR HIGH-DEFINITION MAPS WITH OPENDRIVE FORMAT UTILIZING MOBILE MAPPING MEASUREMENTS

Hao-Yu Pai^{1*}, Jhih-CingZeng², Meng-Lun Tsai³, Kai-Wei Cheng⁴

¹ NCKU, Department of Geomatics, No. 1, Daxue Rd., East Dist., Tainan City 701401, Taiwan - handsomepail@gmail.com ² NCKU, Department of Geomatics, No. 1, Daxue Rd., East Dist., Tainan City 701401, Taiwan -

cing850831@geomatics.ncku.edu.tw

³ NCKU, Department of Geomatics, No. 1, Daxue Rd., East Dist., Tainan City 701401, Taiwan -

taurusbryant@geomatics.ncku.edu.tw

⁴ NCKU, Department of Geomatics, No. 1, Daxue Rd., East Dist., Tainan City 701401, Taiwan - kwchiang@geomatics.ncku.edu.tw

Commission I, WG 1/7

KEY WORDS: OpenDRIVE, Extraction of Road Factors, Lane Lines, Point Cloud, Automated Modelling, HD Maps.

ABSTRACT:

High-definition maps (HD Maps) becomes a trend supporting autonomous vehicles (AVs) which provides accurate auxiliary information about geometries of road, such as center lines of lanes, geometries of roads, traffic signs, etc. It is not restricted by the severe environment, lack of Global Navigation Satellite System (GNSS) signal, or torrential rain, for Avs; furthermore, the standard of HD Maps is defined as tens of centimeters positioning level. However, the production cost of HD Maps is enormous involving human resources and time. The general way of producing HD Maps is to vectorize the roads from point clouds to shapefiles and import them into Geographic Information System (GIS) software to generate HD Maps. In order to simplify this complicated process, this article purposes an algorithm to automated generate high-definition maps from road marks' data which is extracted from point clouds. The methodology in this article is mainly dived into three steps, extraction of road marks, classify lane lines, and modelling OpenDRIVE files. The achieved 2D and 3D accuracies of proposed algorithm in first fields are about 0.069 m in 2D and 0.107 m, respectively.

1. BACKGROUND

The usual maps, like Google Map, are interpreted for humans and not for AVs. The issue that information can be transferred to be recognized by Avs and do planning as the same way humans do is the primary task for high-definition maps (Edgar, 2021). Nowadays there are many high-definition map formats, and each of them provides different descriptions about traffic information. The main formats are Autoware Vector Map, OpenDRIVE, Lanelet2, and Navigation Data Standard (NDS). Autoware Vector Map comprises lots of thematical csv files, and there are 32 types of csv files about the different road information distributed in Vector Map (Edgar, 2021). OpenDRIVE is mainly used for simulation applications, and it is an open format organized in a hierarchical structure and compiled in an XML format. The format describes true road geometry and contains one reference line and lanes which are built around that (Edgar, 2021). NDS is the standard between automotive companies, map data producer, and manufacturers of navigation devices, but it is not a full open format (Edgar, 2021). The last format, Lanelet2, is a map that uses six different primitive types to describe all the map data, moreover, it is not just for some isolated applications like localization or motion planning, but for all potential tasks that ADs could need from HD maps (Edgar, 2021). The comparison within these four formats is shown as Table 1, and it scores for each one based on different aspects (Edgar, 2021). In Table 1, each format is not good at the same aspects, and there are three formats with high scores compared with each other, NDS, Lanelet2, and OpenDRIVE, respectively.

Criteria	OpenD RIVE	Autoware Vector Map	Lanel et2	ND S
Creation	1	3	4	2
Applications	4	3	3	4
Adoption	4	1	2	5
Expressiveness	4	2	3	5
Accessibility	4	2	5	1
Total	17	11	17	17
Accessibility Total	4 17	2 11	5 17	1 17

Table 1. Comparison within different aspects of each HD Mapformat (Edgar, 2021).

1.1 Navigation Data Standard

According to (Salvatore, 2020) and the official website of NDS, NDS is a standardized format for automotive-grade navigation database, but it is not open. Since it requires a purchase of a license, this format is not ideal for academic research. It is illustrated in Table 1 that NDS performs excellent in adoption and expressiveness owing to the essence of NDS. It is originally created for major automotive companies, so that it is the most integrated format of all. However, due to low accessibility of NDS, it is hard to be widely utilized for the ordinary people. People must purchase a license of this format, and this requirement forces people to take another one as the standard format.

1.2 Lanelet2

Lanelet2 is one of HD Maps formats with high accessibility as mentioned in (Salvatore, 2020). It is recommended to use the Lanelet2 format in any case for two simple reasons, it is an open-

^{*} Corresponding author, handsomepail@gmail.com

source format and possible to maintain more information. It describes the roads based on the factors, atomic lane sections, and consists of different layers (Fabian et al., 2018). Lanelets, the atomic lane sections, contains the relationship with each other, and this description is simple to present the topology of lanes. In (Salvatore, 2020), Lanelet2 divides the world into a hierarchical structure of six different primitives including Points, linestrings, polygons, lanelets, areas and regulatory elements, respectively. It is constructed in six layers to describe different types of information individually. It is efficient to save abundant traffic information hierarchically. Although it seems that Lanelet2 is the most suitable for people, it has not been developed for a long while as another HD Maps format. There are some issues of adoption that need to be solved, so that it is not the best choice in this article.

2. INTRODUCTION

2.1 OpenDRIVE

Besides NDS and Lanelet2, when it comes to a practical HD Map format, OpenDRIVE is the common one. As one of the common HD Maps formats, OpenDRIVE is good at application, adoption, expressiveness, and accessibility these four concepts, as shown in Table 1. As long as people can deal with the problem of creation which is mentioned in abstract paragraph, it will absolutely be the most common HD Map format. Moreover, it is imperative to unify a standard format for the convenience of exchanging HD Maps. According to (Zenzic, 2020), it recommends adopting OpenDRIVE as the standard HD Map format.

OpenDRIVE is explicitly designed to describe the physical relationship of road networks and relative road objects. It emphasizes on the geometries of the reference lines and the topologic of lanes in the main body, as for road objects, such as traffic signs, barriers, and sidewalks, they are defined individually detailly in different parts. Also, it allows people checking and editing simply, and enhances the interoperability. One of the advantages is that it can be easily converted to other HD Map formats like Vector Map, Lanlet2 and so forth (Hatem et al., 2019). With these benefits and recommendation, OpenDRIVE is the most appropriate and promising HD Map formats for this article.

2.2 Structure

The structure of OpenDRIVE format is illustrated in Figure 1. OpenDRIVE is compiled with XML format, and everyone can edit or even create a fictious HD Map in this format. Header, road elements, controller, junction, and station construct (Barsi et al., 2021) OpenDRIVE format as below. This figure illustrates the predefined hierarchy of each main component in OpenDRIVE (OpenDRIVE, 2021). This study focuses on the establishment of the road segment, and it only remains planView, elevationProfile, and lanes parts without link (connection), lateral profile and signals, respectively.



Figure 1. The structure of OpenDRIVE format (Barsi et al., 2021).

The first component, header, describes the basic information, such as production date, name of the database and companies, the version of OpenDRIVE, the projection information (Barsi et al., 2021). These descriptions are for those users who want to know the background information of this production, and they do not affect the production too much. They can be edited casually depending on the developers' methods.

Road, the majority of this article, mainly describes roads' structure that is divided into three parts, reference line, lanes, and features. OpenDRIVE is basically constructed by reference lines, and it presents the shape of roads by the geometries of reference lines. The reference line is in a specific coordinate, reference line system (s/t coordinate), and s coordinate means the distance along with the reference line in meters. As for t-coordinate, it defines which lane and which side it is. For example, if t-coordinate is negative, the position is at the right lane.

There are four kinds of geometries simulating a reference line to fit various road scenes, line, spiral, arc, paramPoly3 (parametric cubic curves), and the other one, poly3 (cubic polynomial), as illustrated in (OpenDRIVE., 2021). The details of each geometry would be introduced later in methodology section. Another addressed issue in road part is the elevation as OpenDRIVE takes a cubic polynomial to fit the elevation of reference line. It remains some residuals in cubic polynomial, hence, it cannot represent the true vertical information.

For lane issue, it includes width, id and attribute of lanes, but we ignore the connection of lanes here. The lane id is based on tcoordinate, and the definition is as same as the t-coordinate. The lane id is an integer, and the shorter distance between reference line and the lane is, the less absolute value of lane id is. When it comes to lane width, it is also presented as cubic polynomial. The result might remain more residuals because of the diverse lane shapes, therefore, the lane will be segmented into lane sections. This strategy can decline the residuals in distinct lane shape within one lane.

3. ROAD MARKS EXTRACTION

The main purpose of this study is to generate high-definition maps of roads from point clouds data, wherefore the result of lane lines extraction would determine the accuracy of the final outcome. The first of all is to obtain the road region, so that it needs to keep ground points and find out road surface. There are some methods achieving this goal, and the mainstream of road surface extraction is based on curb structure, which supplies height jump between sidewalks and roads (Yang et al., 2013). A method to distinguish road surface regions based on non-ground point removal and road edge detection is also proposed in (Zhao, 2017). In non-ground point removal, voxel-based upward growing algorithm is used to define non-ground points and remove them. In road edge detection, it segments data, detects curb and finally refines road edge by RANSAC to get road surface. However, there is not any curb structure in the experiment field of this article which is shown in Figure 2. It is restricted to adopt curb as a road extraction condition, so the alternative ways of extraction are applied instead. A method to overcome the issue of the environment without curb structure is developed in (Yadav et al., 2018). There are three steps to extract road surface, Local Road Patches Extraction, GPBM based Local Road Surface Model Estimation, and False Detections Filtering Based on Clustering. According to (Zeng, 2020), it takes the difference of intensity between roads and sidewalks to extract road edges, yet there are some mistakes at the boundaries with dirt.

The methodology in this article improves the performance at the edges by region growing algorithm, and it is illustrated in Figure 3 with the flowchart. In the flowcharts, the blue boxes represent the imported data, the green boxes mean the used algorithm, the yellow boxes are the results after one phase, and the final result is presented as the orange box.



Figure 2. Points cloud data in the experiment field without curb structure.

3.1 Algorithm

Point clouds data is imported as the unprocessed data, and trajectory is taken as the supporting data. In order to reduce the data processing capacity, coordinate translation from absolute coordinate to local coordinate and down-sampling are chosen. After pre-processing, Cloth Simulation Filter (CSF) is introduced to obtain the ground point clouds in the algorithm. It can simulate a ground surface to filter out the non-ground points without a certain height threshold. The remaining data, ground points, is divided into low and high intensity datasets by the intensity filter. In high-intensity dataset, it retains point clouds of sidewalks, so that it cannot recognize the road mark from high intensity dataset only by the intensity filter.

The primary task is to remove point clouds of sidewalks if we want to extract road marks. Obviously, it is essential to obtain road edges to get road surface and eliminate sidewalks. Owing to the lacks of curb structure in this experiment field, region growing is used for clustering road surface to obtain road edges. In low intensity dataset which the road surface belongs to, the algorithm takes height value as the condition to do region growing. After the extraction of road edges, the region of road surface is obtained. Next, the algorithm takes Statistical Outlier Removal (SOR) filter to refine the result and imports the simulated ground surface from CSF to find out the point cloud of road marks. The point clouds of these road marks are disorder, so they are classified into stop lines, arrows, crosswalks, and lane lines by Euclidean distance and oriented bounding box (OBB) classification with the different geometry features between each road mark. For example, the lane lines should contain a long, straight line, and crosswalks should be a set of parallel lines. In this study, only the production of lane lines is addressed, so the point clouds of lane lines are reserved and refined to be imported in the modelling algorithm.



Figure 3. The flowchart of extraction algorithm.

4. POLYLINES MODELLING

According to (Chaiwat et al., 2010), a cubic spline can be fitted to a series of points, the interpolations of data. It provides a useful way of approximating a smooth Equation (1), it is reasonable enough to adopt cubic spline fitting for modelling.

$$f_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i,$$
(1)

After importing refined lane lines point clouds and removing noise, the algorithm takes cubic spline fitting to generate lane lines with the distance about 10 cm among each point. The algorithm of modelling polylines depicted in Figure 4Figure 4. The flowchart of lane line modelling algorithm. is able to automatically model lane lines of each road, and the following introduction involves the classification for extraction data and the assessment for the results.



Figure 4. The flowchart of lane line modelling algorithm.

4.1 Pre-Processing (Segment Extracted Dataset and Trajectory)

The main purpose of modelling lane lines is to create the factors of OpenDRIVE, like the reference line, lane width, or length of roads. Before modelling, the extracted data and the trajectory have to be divided into different roads. With these data pre-processing, the algorithm, which is mentioned in the following content, can not only model every lane line of each

road, but also avoid wrong trajectory points detecting. It firstly takes Density-based spatial clustering of applications with noise (DBSCAN) to cluster extracted datasets into many roads. Due to the characteristic of lane lines distribution, there is not any lane line in the junction. It is able to distinguish different roads and process each road individually.

The requirement of the following algorithm is importing trajectory data as the reference of classification. If the extracted dataset detects wrong trajectory points which does not belong to the same road, the result of classification will be destroyed. As the consequence, there is necessary to segment the trajectory data into different roads as the result of after clustering extracted dataset. The algorithm adopts convex hull to get the boundary of every road clusters. With these ranges, the algorithm can find out which trajectory points are involved in the same road as the extracted data.

4.2 Classify Extracted Dataset of One Road

The first step of modelling the polylines after pre-processing is classifying these lane lines extractions. These datasets have not been defined with lane attributes to determine which lanes they belong to.

Model The Far-Right Lane Line: Consequently, the 4.2.1 lane lines extractions are firstly translated into reference line system, which the original point is the first point of each segmental trajectory, therefore, the logic of classification is mainly based on the trajectory. In reference line system (s/t coordinate), the algorithm compares mean value of t-coordinate from the first to fifth point of each dataset to search the lane line on the far-right. All datasets are put in the different t-coordinate clusters by DBSCAN. The reason why this algorithm chooses DBSCAN is that we cannot make sure the number of lane lines, and one of the advantages of DBSCAN is that it does not need to define the number of clusters, like K-means clustering. The minimum cluster is set as the lane line on the far-right within these clusters. After we find out which datasets belong to the lane line on the far-right, the algorithm will adopt cubic spline fitting to fit them.

4.2.2 Classify Every Lane Lines: If the lane line on the farright is derived successfully, the remaining datasets can be classified into the lanes they belong to by Euclidean distance. With the logic of traffic, all points of each lane line should contain same Euclidean distance from the lane line on the far right, and DBSCAN is used to classify again. The algorithm cubic fits each lane line with the distance about 10 cm among each point. As for the roads in junction, it has to be modelled by cubic spline with the point of two intersecting lane lines owing to the lacks of lane lines in junctions. Because the road marks extraction data in junctions is not available, so that the modelled lane lines are not as accurate as the general roads. The result will be verified with the HD Maps which have been verified by mapping companies.

The algorithm calculates the shortest distance by Euclidean distance from each point of the results to the verified HD Maps to analyze the result of modelling the polylines.

5. MODEL THE ROADS IN OPENDRIVE FORMAT

In this part, it illustrates how to model the polylines data into OpenDRIVE format. As mentioned in (Han, 2011), separating the road into several segments by a reasonable way, it is dived into four kinds of geometries, Line, Spiral, Arc, Polynomials, which are employed in OpenDRIVE. Moreover, the identification of each geometry in this article is as same as the paper (Han, 2011). which is based on the feature of curvature. The main differences between this article and the paper are measurement data and the depiction of polynomial function. The measurement data in (Han, 2011). is obtained from OpenStreetMap data source, and it only retains horizontal information. As the result, it cannot implement three-dimensional information for the elevationProfile and crossfall parts in OpenDRIVE. In addition, the algorithm in this article replaces the cubic polynomial function because of the OpenDRIVE version update. In Figure 5, it illustrates the flowchart of modelling roads in OpenDRIVE format algorithm, and the detail will be introduced below.



Figure 5. The flowchart of generating OpenDRIVE algorithm.

5.1 Classify Lane Attribute

The most important factor in OpenDRIVE is the reference line, and it defines a road's structure. The priority is to define which lane line belongs to the reference line from every lane lines. The attribute of reference line belongs to the center lane line, and it is defined as the place where the direction of trajectory changes, or the divisional island lane where trajectory does not cover. The algorithm requires importing the trajectory data which covers all lanes. Being restricted by the traffic rules, we can ensure that the direction of trajectory is logical. On account of the definition of the center lane line in the algorithm, the direction changes or the lane without trajectory data, we are able to find the reference line by the trajectory data of all lanes. However, there is an exception, motorcycle lane, and it has to be excluded. The divisional island lane and the motorcycle lane belong to the lane which does not contain trajectory data. Since divisional islands locate at the middle of the road, the difference between motorcycle lanes and divisional islands is the position. As these conditions, we can automatically define which lane line is the reference line.

5.2 Geometry Parameters of Reference Lines

The primary of this part is to obtain each curvature of reference lines and the defined geometries. The functions of calculating curvatures and the interval of i and j are shown as below, Equation (2) and (3). Curv(i) means the curvature of every three points, and curv(j) means the mean value of every three curvatures. The meanings of averaging in Equation (3) are to blur the effect of outliers and decrease computation without losing characteristic of reference lines by the policy of taking three curvatures.

$$curv(i) = \frac{\ddot{\vec{\gamma}} \times \dot{\vec{\gamma}}}{|\vec{\gamma}|^3} = \frac{x''y' - x'y''}{(x'^2 + {y'}^2)^{\frac{3}{2}}}$$
(2)

$$curv(j) = \frac{\sum_{n=1}^{3} curv(4 \times j - n)}{3}$$
(3)

where i = 1: (numbers of points in reference line) - 2 j = 1: ($i \div 4$)

As long as the curvatures are calculated, the reference lines are segmented into four geometries by the features of curvatures. The following introduction will individually describe the characteristics and the XML structure of each geometry. The jointly owned geometry parameters are s-coordinate (m) of start position, start position of x (m) and y (m) in inertial coordinate, start orientation (hdg) (rad) in inertial coordinate, and the length (m) of segmented reference lines (OpenDRIVE, 2021). The scoordinate is calculated from the sum of the length of previous geometries, and the first one is zero. The x and y coordinate are the position of the first point, and the hdr is obtained from the arctangent value of the first two points in inertial coordinate.

5.2.1 Line: It is a linear geometry, and the curvature of Line is zero. It may lose the characteristic if a segment of reference lines is presented as Line in OpenDRIVE format without any evidence. As the consequence, it has to be strictly examined if the segment belongs to Line part, and it is presented as below for example. The threshold of curvature in Line is 0.001, which means that it will be seen as Line if the curvature value is lower than 0.001. The length of each curvature is about 0.4 (m) and the radius of which is classified in Line is 1000 (m) at least, therefore, this threshold is critical enough to rely on due to the huge disparity.

< planView>
 <geometry s= "0" x= "1" y= "2" hdg= "0.3" length= "4">
 </geometry>
 </planView>

5.2.2 Spiral: Spiral is a clothoid geometry that describes a part of reference lines with a linear curvature, which means that the $\Delta curvature$ in Equation (4) is a constant. It is characterized by the curvatures of the first point (curvStart) and the last point (curvEnd).

$$\Delta curvature = curvature(j+1) - curvature(j) \quad (4)$$

< planView>

<geometry s= "0" x= "1" y= "2" hdg= "0.3" length= "4"> <spiral curvStart="0.0" curvEnd="0.013"/> </geometry> </planView>

5.2.3 Arc: Arc is a geometry with a constant curvature. The reason why we do not take Arc to present all segments of reference lines is that the file will be too large to conveniently transfer if we present all curvatures. Furthermore, it is not necessary achieving the extremely high accuracy with about 0.1 cm of the final result.

```
< planView>
  <geometry s= "0" x= "1" y= "2" hdg= "0.3" length= "4">
        <arc curvature="0.013"/>
        </geometry>
  </planView>
```

5.2.4 Polynomial: Polynomial is a geometry which uses parametric cubic curves to fit a part of reference line. If there is not any regulation of curvatures, the segment will belong to Polynomial. The structure of Polynomial in OpenDRIVE is presented as below.

The first step of generating roads in Polynomial is converting the data from inertial coordinate (x/y coordinate) to local coordinate (u/v coordinate). U-coordinate is the axis along the start point of the reference line, and v-coordinate is the axis of lateral deviation from the first point of the reference line. According to the OpenDRIVE V1.6.1 User Guide in 2021, Generating roads with parametric cubic curves only require xand y-coordinates (OpenDRIVE, 2021). For reasons of consistency to cubic polynomials, they may be calculated simultaneously to cubic polynomials using local u- and vcoordinates, it cubic fits the u- and v-coordinate with interpolation parameter p which is in the range from 0 to 1. The concept of a parametric cubic polynomial expressed as Equation (5) and (6) is illustrated in Figure 6.

< planView>

<geometry s= "0" x= "1" y= "2" hdg= "0.3" length= "4">
 du="0.000" bU="1.000" cU="-4.648e-09"
 du="-2.6297e-08" aV="0.0000" bV="1.66538e-16" cV=" 1.98778e-04" dV="-1.3159e-09" pRange="normalized">

 $u(p) = aU + bU \times p + cU \times p2 + dU \times p3$ (5)

 $v(p) = aV + bV \times p + cV \times p2 + dV \times p3$ (6)



Figure 6. A sample of parametric cubic polynomial. (OpenDRIVE V1.6.1)

5.3 Elevation

After obtaining parameters of each geometry, there is a parameter that has not been calculated, elevationProfile. In OpenDRIVE format, elevationProfile is presented as a cubic polynomial that fits all elevation value from the start point to the end point. It is shown as below, and the explanation, "a" represents the constant term and "d" represents the cubic term, is widely used in OpenDRIVE format, if there is one parameter presented by cubic polynomial.

<elevationProfile>

<elevation a="44.70" b="0.0016" c="0.00" d="-0.00" s="0"/> </elevationProfile>

5.4 Length of Each Geometry

In length part of OpenDRIVE format, it is dived into three functions, and each geometry owns a specific length function.

5.4.1 Line: As we know that the distance of a line can be calculated by Euclidean distance from the first point to the last point of a Line geometry. It is shown as Equation (7).

$$Length_{L} = \sqrt{(x_{n} - x_{1})^{2} + (y_{n} - y_{1})^{2}}$$
(7)

5.4.2 Spiral and Polynomial: There is not certain function for Polynomial and Spiral, so the length of them is calculated by the sum of distance within each interpolation in this article.

5.4.3 Arc: The function of circumference is utilized in Arc, and the radius of arc (R) is the reciprocal of curvature. It is shown as Equation (8), where "L" is the Euclidean distance from the start point to the end point in an Arc geometry.

$$Length_{R} = 2 \times \pi \times \frac{1}{curvature} \times \frac{\sin^{-1}(\frac{L}{R}) \times 2}{360^{\circ}}$$
(8)

5.5 Attributes in Lanes

In Lanes part of OpenDRIVE format which is shown as below, there are two major attributes that determine what a lane is in this article, id, which defines the lane is at left or right side, and lane width. The first mission is to define correct identifications following the logic in OpenDRIVE format which has been mentioned in the introduction paragraph. The methodology is almost as same as the classification in modelling the polylines from extraction, and it is also required to translate from x/y coordinate to s/t coordinate. After coordinate translation, the algorithm compares t-coordinate of first point of each lane line firstly, and gives the relative id for each lane. The relative id depends on the direction of reference line, and we take on the farright lane as example. If a road is a two-way road with two lanes for each direction, the id of on the far-right lane which tcoordinate is the minimum will be defined as -2; on the other hand, the id of the most-left lane which t-coordinate is the maximum will be defined as 2. As for the lane width in OpenDRIVE format, it is also a cubic polynomial, and it cubic fits the shortest distance of all points from reference lines to lane lines. Thus, it might ignore some features of lane width, and it is crucial to decide whether it has a necessity to separate lanes into different lane sections.

```
<lanes>
  <laneSection s="0">
    <left>
       <lane id="1" type="driving" level="false">
         <width a="3.625" b="-0.0103" c="0.000" d="-0.000"
         sOffset="0"/>
       </lane>
     </left>
     <center>
       <lane id="0" type="none" level="false"/>
    </center>
    <right>
       <lane id="-1" type="driving" level="false">
         <width a="3.625" b="-0.0103" c="0.000" d="-0.000"
         sOffset="0"/>
       </lane>
    </right>
  </laneSection>
</lanes>
```

6. EXPERIMENT

The experiment field in this study is Taiwan CAR (Connected, Autonomous, Road-test) Lab, Tainan, Taiwan, and it is the official test field for autonomous vehicle in Taiwan. It retains thirteen traffic scenarios, and test field are chosen as red part, which is illustrated in Figure 7. Every section contains different characteristic, and the policy of selection is checking if the algorithm works for various roads or not. Due to the limitation of test field that there are some missing of lane lines in the reality, the extraction data will lose these observations.



Figure 7. The satellite image of Taiwan CAR Lab (from Google Map). The red mark parts are the road sections that is chosen as the test data in this article

The equipped sensors of the commercial MLS system are an GNSS receiver, an IMU, cameras, laser scanners, a Distance Measurement Instrument (DMI), and a ladybug. The specifications in Table 2 and the sensors on the mobile vehicle are shown as below in Figure 8.

GNSS/IMU/DMI		
Positioning (absolute)	20~50 (mm)	
Positioning (relative)	10 (mm)	
Scanners_VQ250		
Laser Pulse Rate (PRR)	60 (kHz)(Max.)	
Scan Rate	Up to 200 LPS	
Accuracy	10 (mm)	
Precision	5 (mm)	
Cameras-CS6		
Resolution	2452×2056	
Pixel Size	$3.45 \times 3.45 \ (\mu m^2)$	
Field of View	80° ×65°	
Exposure	38 (µs~60s)	

Table 2. Specification of sensors.



Figure 8. The equipment of commercial MLS system.

Since the previous trajectory which is from the mapping company is a little wired, some sections of the trajectory are on the sidewalks, we observe it again by the higher quality sensors. The equipped sensors are iMAR iNAV-RQH-10018 which specification is shown as below in Table 3 as the IMU and NovAtel PwrPak 7D-E2 as the GNSS receiver. iMAR iNAV-RQH-10018 is a navigation-grade INS, and the precision of observation is great enough to rely on it. We use the software, NovAtel Inertial Explorer®, to obtain a new and reliable trajectory data.

	Gyroscope	Accelerometer
Bias Instability	<0.002°/hr	$< 10 \mu g$
Random Walk Noise	<0.0015°/hr	$<8 \mu g/\sqrt{Hz}$

Table 4. The specification	n of iMAR	iNAV-RQI	H-10018
----------------------------	-----------	----------	---------

7. RESULT AND DISCUSSIONS

In this section, the modelled polylines with proposed algorithm in OpenDRIVE format will be evaluated with the existing Taiwan HD Map which is produced by the mapping company and certified by the high-definition map research center in Taiwan. The comparing result is shown as below in Table 4 and the methodology of comparison is calculating the shortest distance from all points of the modelling polylines to all points of the reference data which is from the mapping company. The company models the lane lines manually from the point cloud data, and generates the shapefiles and converts them to CSV files as the reference data.

	2D	3D
RMSE	0.045 (m)	0.062 (m)
Average error	0.040 (m)	0.055 (m)
STD	0.021 (m)	0.027 (m)

Table 5. The results of modelling polylines.

According to Taiwan HD Map standard (High Definition Maps Research Center, 2020), the accuracy in 2D should be smaller than 20 cm, and 30 cm in 3D, respectively. We can see that all indicators fulfill the standard for HD Maps requirements in each dimension. The RMSE is taken as the index to present how the algorithm performs, and they are 4.5 cm in 2D and 6.2 cm in 3D, respectively.

The Figure 9 and Figure 10 illustrate how the result looks like when it is imported into different platforms, such as OpenDRIVE Online Viewer, Google Earth, and CAR Learning to Act (CARLA) simulator. Moreover, we can check if it is available to be widely utilized or not. Figure 9 is in OpenDRIVE Online Viewer, a free HD Maps online viewing tool which is only for OpenDRIVE format launched officially. It allows people opening the files with .xodr file extension directly, and displays the files in 3D model. We can see that all roads are clearly displayed in OpenDRIVE Online Viewer.



Figure 9. The result in OpenDRIVE Viewer.

Figure 10 illustrates the converted elements in the Google Earth, but Google Earth only supports the kml file. In order to import the result in Google Earth, the ASSURE Mapping tool is applied to transform the result from OpenDRIVE format to Google Earth Map which is compiled in kml format. Since Google earth map does not display lane width, the converted result only remains polylines that present center lines. After conversion of the HD Map, the center line of each lane is almost fit the layer of the satellite image.



Figure 10. The converted result in Google Earth (from Google Earth).

In order to compare the result after modelling the roads in OpenDRIVE format with the reference data, it has to be converted to the Asian vector map as a CSV file to present in points. The ASSURE Maps which allows people converting HD Maps format is adopted to convert the HD Maps from OpenDRIVE to the Asian vector map. The comparison is illustrated in Table 5.

	2D	3D
RMSE	0.045 (m)	0.062 (m)
Average error	0.040 (m)	0.055 (m)
STD	0.021 (m)	0.027 (m)
Table 6 The regults of modelling roads in OpenDRIVE		

Table 6. The results of modelling roads in OpenDRIVE format.

As the table shown above, all the indicators fulfill the accuracy requirements of the Taiwan HD Map standard, and it is reasonable that the performance in 2D is better than 3D. The reasons why there is a little error between the result of modelling polylines and roads in OpenDRIVE format are that the methodology of obtaining mean curvature exists some lost and the geometry parameters cannot present roads without error. Besides, it compares the result after map translation with the shape files from the mapping company, and it exists some distortion in this step. However, it is still reliable enough that the RMSE is 6.9 cm in 2D and 7.9 cm in 3D. It is a dependable result that automatically modelled roads in OpenDRIVE format are under the HD Maps standard.

8. CONCLUSION AND FUTURE WORK

The importance of high-definition maps is so significant that it will become a trend supporting autonomous vehicles and improve the result of navigation. However, it exists enormous production cost in generating an HD Map by manually, and it is in a predicament that there is no automatically generating method extensively applicated. In this article, it proposes an algorithm that can automatically generate HD Maps in OpenDRIVE format for roads with multiple lanes. It includes road elements extraction and lane lines modelling. In extraction part, the algorithm adopts CSF and region growing-based road edge extraction to get road marks in high intensity dataset which is classified by intensity filter. In modelling part, the algorithm firstly classifies the extraction dataset into different lane lines and models them together by cubic spline fitting. After that, it extracts the geometry parameters for OpenDRIVE format to complete HD Maps. Besides, the result of the algorithm is reliable enough under the Taiwan HD Map standard. The accuracies of the results are 6.9 cm in 2D and 7.9 cm in 3D, respectively. The algorithm reduces the time of HD maps production. Moreover, simply by ASSURE Maps the outcome of the algorithm can be transformed to other HD Map formats to obtain more general interchangeability (Hatem et al., 2019).

ACKNOWLEDGEMENTS

The author sincerely appreciates the technical assistance from the senior research associate in the High Definition Maps Research Center, Ms. Jhih-Cing Zeng and the professional advices from the supervisor, Dr. Kai-Wei Chiang. Besides, the author also wants to acknowledge Dr. Hatem. The author cannot finish the research without Dr. Hatem's professional assistance about ASSURE mapping tool. The MOI within Executive Yuan in Taiwan is acknowledged for providing financial support for this study. This study cannot be completed without their assistance

REFERENCES

Bauer, S., Alkhorshid, Y., Wanielik, G., 2016: Using highdefinition maps for precise urban vehicle localization, *Proceedings of 19th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 492-497.

Chaiwat, N., Theekapun, C., 2010: Lane detection using smoothing spline, *3rd International Congress on Image and Signal Processing*.

Chiang, K. W., Tsai, G. J., Zeng, J. C., 2021: Mobile Mapping Technologies, *Urban Informatics, pp. 439-465*.

Edgar, S., 2021: Creating High-Definition Vector Maps for Autonomous Driving, *Masters' Thesis, 15 European Credit Transfer and Accumulation System (ECTS)*, pp. 27-28.

Fabian, P., Jan-Hendrik, P., Johannes, J., Stefan, O., Maximilian, N., Florian Kuhnt, Matthias, M., 2018: Lanelet2: A high-definition map framework for the future of automated driving,

21st International Conference on Intelligent Transportation Systems (ITSC).

Han, S., 2011: Automatic Generation of OpenDrive Roads from Road Measurements, *Chalmers University of Technology University of Gothenburg, Department of Computer Science and Engineering, SE-412 96 Göteborg, Sweden.*

Hatem, D., Nicky, K., Aladdin, A., 2019: ASSURE Maps.

High Definition Maps Research Center, 2020: HD Maps data contents and formats standard v1.1.

M. Barsi, and A. Barsi., 2021: Building OpenDRIVE Model from Mobile Mapping Data, The International Archives of the Photogrammetry, *Remote Sensing and Spatial Information Sciences, Volume XLIII-B4-2021, XXIV ISPRS Congress (2021 edition).*

Navigation Data Standard., 2020: The worldwide standard for map data in automotive eco-systems., *Retrieved from https://nds-association.org/*.

OpenDRIVE., 2021: ASAM OpenDRIVE V1.6.1 User Guide, *Retrieved from http://www.opendrive.org/*.

Salvatore, F., S., 2020: A COMPLETE END-TO-END SIMULATION FLOW FOR AUTONOMOUS DRIVING FRAMEWORKS, Politecnico di Torino, Department of Electronic and Telecommunications, Master's Degree Course in Mechatronic Engineering, Final Project Work.

Tsai, G. J., 2020: Seamless Navigation and Mapping Using INS/GNSS/LiDAR SLAM Multi-Fusion Schemes, *Ph.D. Thesis, National Cheng Kung University, Tainan, Taiwan.*

Wang, H., Wang, C., Chen, Y., Yang, W., Li, J., 2014: Extracting road surface from mobile laser scanning point clouds in large scale urban environment, *Proceedings of 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 2912-2917.*

Yadav, M., Lohani, B., Singh, A. K., 2018: Road Surface Detection from Mobile LiDAR Data, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-5.*

Yang, B., Fang, L. and 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.

Zeng, J. C., 2020 : Automated Road-Elements Modelling and Centerline Generation for High-Definition Maps Utilizing 3D Point Cloud, *National Cheng Kung University, Tainan, Taiwan*.

Zenzic., 2020: Geodata report-analysis and recommendations for self-driving vehicle testing Zenzic consultation findings Geodata report-analysis and recommendations for self-driving vehicle testing, *Zenzic consultation findings*.

Zhao, H., 2017: Recognizing Features in Mobile Laser Scanning Point Clouds Towards 3D High-Definition Road Maps for Autonomous Vehicles, Master's thesis, University of Waterloo, Ontario, Canada.