Mapping the Future: High-Definition Map Generation for Self-Driving Cars

Teckhua Chiang

State University of New York at Buffalo

September 2018

ABSTRACT

High-definition maps are used by self-driving cars to navigate through complex traffic environments. Even though high-definition maps are a critical element of existing autonomous vehicle algorithms, generating them is a challenge due to the significant cost of purchasing the necessary sensors and the limited availability of procedures for turning data into maps. Thus, the difficulty of high-definition map generation impedes self-driving car development. This research makes it possible to quickly and efficiently create high-definition maps because it produced an innovative generation procedure that requires less sensor data input compared to traditional generation methods, utilizes open-source software, outputs maps in a standardized configuration, and can be modified with alternative mapping algorithms. This procedure integrates data collection, preparation, processing, merging, and annotation, thereby expediting the task of translating sensor data into high-definition maps. As a result, this process greatly reduces the barriers to self-driving car research and accelerates the realization of a future with autonomous transportation.

Keywords: high-definition maps, self-driving cars, autonomous vehicles

Researched under the supervision of Professor Chunming Qiao, SUNY Distinguished Professor, Department Chair, Adjunct Professor of Electrical Engineering, and Roman Dmowski, iCAVE2 Project Manager

I. INTRODUCTION

High-definition maps, otherwise known as HD maps, are geospatial maps built for robotics applications. Unlike conventional maps, which only provide a rough approximation of a region, high-definition maps are exceptionally precise in their description of an area, often attaining centimeter-level precision. Due to the fact that computers require a precise understanding of 3D spaces to maneuver, high-definition maps are vital for robotics development and research.

Especially in the case of autonomous vehicles, high-definition maps play a critical role because their precision and comprehensiveness aids navigation in self-driving cars. Due to the complex interactions and unpredictable conditions that occur in everyday driving situations, real-time decision-making is difficult to implement reliably without using prior information to aid autonomous algorithms. The precision of high-definition maps helps autonomous vehicles localize themselves by comparing scans of the current environment with their existing record of the area, allowing them to identify exactly where they are positioned within several centimeters of error. Meanwhile, the comprehensiveness of HD maps also reduces the computational burden on autonomous software by alerting perception systems to potential points of interest, such as the location of lane markings, street signs, and traffic lights, for more efficient recognition. With regard to autonomous vehicle research, high-definition maps are required to run and test the majority of self-driving algorithms, thus necessitating a method of easily generating such maps.

While high-definition maps are an essential component of existing autonomous vehicle architectures, generating them is a challenging and time-consuming task. Traditional high-definition maps are generated using data from a variety of sensor systems, including camera, ultrasound, radar, LiDAR (Light Detection and Ranging), GPS (Global Positioning System), and IMU (Inertial Measurement Unit). However, the cost of purchasing all of these sensor systems and integrating them into a single vehicle platform often makes the process of obtaining the necessary data incredibly restrictive. Furthermore, there is no easily-accessible open-source algorithm for processing these sensor inputs into a usable high-definition map. As a result, researchers are forced to either outsource high-definition map generation to professional mapping companies or develop their own mapping algorithms, both of which are not conducive to quick and flexible

experimentation. Finally, there is no industry standard for a high-definition map format that is portable across multiple autonomous vehicle software systems. Thus, the difficulty of high-definition map generation poses a significant barrier to future autonomous vehicle research.

This paper explains how to use open-source software, such as Autoware [1], an ROS-based [2] software for self-driving vehicles, in order to quickly and efficiently generate high-definition maps in a widely-adopted data format using LiDAR input. It provides a step-by-step guide on how to transform raw sensor data into full high-definition maps, as well as notes on possible issues that may be encountered along the way. Finally, it discusses the positive and negative aspects of high-definition map generation, as well as potential areas of research in the future.

II. BACKGROUND

This research was part of the iCAVE2 Project [3] led by Professor Chunming Qiao and conducted at the University at Buffalo. The project aims to develop an integrated 5-in-1 instrument for Connected and Autonomous Vehicle Evaluation and Experimentation (iCAVE2). The instrument consists of five components: a driving simulator, a traffic simulator, a network simulator (NS), several instrumented vehicles (IVs) including an autonomous vehicle (AV), and an instrumented environment in SUNY Buffalo's Motion Simulation Lab [4]. The iCAVE2 simulator is intended to provide an unprecedented virtual model for testing autonomous vehicle algorithms and systems. It combines the benefits of existing simulators and road testing facilities by providing a flexible, scalable, low-cost, realistic, and most importantly, safe, platform for the evaluation of connected vehicle and autonomous vehicle technologies. Thus, it is particularly well-suited for researching topics such as safety, efficiency, and sustainability with regard to experimental autonomous technologies and rare road conditions.

This investigation of high-definition map generation was prompted by the iCAVE2 project's additional goal of activating and experimenting with the autonomous mode for a self-driving car running the Autoware and Apollo [5] software platforms. By discovering a fast and efficient method of high-definition map generation, it is now possible to test autonomous algorithms in a real-world environment. This provides the iCAVE2 project with a variety of new

avenues for research, including comparing the performance of self-driving software within the iCAVE2 simulator with its performance on a real autonomous vehicle.

III. DEVELOPMENT

The procedure outlined in this paper is based upon extensive research on relevant academic papers, articles, and discussions. After identifying the key elements required for high-definition map generation, further research was conducted in order to evaluate and select the appropriate approach and software for each individual stage. In the end, this procedure integrates data collection, preparation, processing, registration and merging, and annotation, as well as localization simulation testing, into a single, streamlined process. By incorporating these core task into a standard framework for high-definition map generation, this procedure makes it possible to quickly and efficiently create high-definition maps because it requires less sensor data input compared to traditional methods, utilizes open-source software, outputs maps in a standardized configuration, and can be modified with alternative mapping algorithms.

IV. REQUIREMENTS

In this research, the sensor data utilized to generate high-definition maps was collected using a Velodyne VLP-16 LiDAR [6] mounted on a Local Motors Olli self-driving bus [7]. However, any vehicle platform with any LiDAR model can be used for LiDAR data collection. As discussed previously, traditional high-definition maps are constructed using data from various sensor inputs. However, this procedure has been streamlined to only require LiDAR data.

The high-definition map generation protocol described below uses a variety of external software to perform operations on the LiDAR data. For instance, the protocol uses Autoware, an "all-in-one" open-source software for self-driving vehicles based on the Robot Operating System (ROS), in order to perform the majority of the required scan transformations. Autoware is a computationally intensive software, requiring the use of a computer with high-processing power, and is run on the Ubuntu Linux distribution [8]. Autoware also provides a range of sensing,

perception, decision, planning, and actuation capabilities. Specifically, Autoware's localization module can be used to test the effectiveness of the high-definition map at the end of the process.

V. PROCEDURE

This high-definition map generation procedure is composed of five major stages: LiDAR data collection, PCAP preparation, NDT processing, ICP registration and merging, and road logic annotation. At the end, it is possible to test the resulting map in the Autoware localization module. Watch the accompanying video tutorial [9] to see how the corresponding actions appear onscreen.

In this procedure, high-definition maps are composed of point cloud structure data and vector feature data, represented by a PCD file and CSV files respectively. The PCD (Point Cloud Data) file [10] stores a point cloud representation of the mapped environment, which allows autonomous software to perform LiDAR-based localization. Meanwhile, a series of CSV files provides a vector map that represents a set of features inherent to the road, such as lanes, stop lines, traffic lights, and intersections, that aid the function of decision and planning modules.

1) **LiDAR Data Collection** – The first stage is to collect the LiDAR data needed to construct a high-definition map. Drive the data collection vehicle around a designated test site several times, preferably in a closed loop, in order to collect a detailed LiDAR scan of the area. Afterwards, the LiDAR scans can be downloaded as a series of PCAP (Packet Capture) files.

   The 3D LiDAR data in the PCAP files can be visualized using VeloView [11], an application developed by ParaView [12], that can playback pre-recorded data and display detailed information about each scan. Each PCAP file is a series of sensor data packets, which represent a series of point clouds that describe the topology of the sensor's surroundings at a single moment. Played in sequence, the point clouds represent the entire environment, but they cannot be used as a high-definition map yet. The individual point clouds must be concatenated together, transformed to match with each other, and then merged into a single point cloud.

2) **PCAP Preparation** – The second stage is using Wireshark [13] to combine the multiple PCAPs into a single PCAP for easier processing. Wireshark is a free, open-source protocol analyzer that allows users to view and operate on the contents of PCAP files. After uploading all of the individual PCAPs, Wireshark automatically concatenates all of the data packets within the PCAPs. The full data stream can then be exported, resulting in a single PCAP file that contains all of the LiDAR scans in sequential order.

   When combining the PCAPs, it must be ensured that they are in sequential order so that they can be concatenated correctly. Furthermore, it is extremely important that there are no jumps in the position of the data collection vehicle anywhere within the scans (e.g. stopped scanning at one location, moved to another position, and then resumed scanning). The rest of the high-definition map generation process relies on the assumption that the scans are continuous, since sudden jumps in position disrupt the Normal Distributions Transform algorithm.

3) **NDT Processing** – The third stage is using Autoware to process the PCAP file containing all of the recorded LiDAR data and convert it into a PCD file. The Point Cloud Data (PCD) file format is a custom format designed to store 3D point cloud data and created by Point Cloud Library (PCL) [14], an open-source project for point cloud processing. While there are many other point cloud data file formats, PCD offers significant advantages over the rest due to its flexibility and speed. First of all, PCD is widely-used, open-source, and human-readable. Furthermore, it provides the ability to process and store organized point cloud datasets.

   After launching Autoware on the command line, an interface will appear on screen. In the "Sensing" tab, select the applicable LiDAR model under the "LiDARs" category. In the "Computing" tab, activate the "ndt_mapping" module. In terminal, run the command:

```
$ rosrun velodyne_driver velodyne_node _model:=the_model_type
_read_once:=true _pcap:"=/the/pcap/path/data.pcap"
```

   Autoware will now start processing the PCAP and converting it into a PCD. In order to confirm that Autoware is performing the Normal Distributions Transform (NDT) [15] correctly, RViz

[16] can be used to visualize the process. After the RViz screen appears, change the configuration to "ndt_mapping.rviz." The transformed point cloud from the LiDAR scans will appear in the visualizer. Then, follow along as the Autoware "ndt_mapping" module registers each LiDAR scan to its predecessor, allowing it to recreate the path of the data collection vehicle and match each point cloud frame to a three dimensional grid. In doing so, Autoware converts the relative point clouds from the PCAP into the static point cloud of a PCD.

After the entire process has completed, return to the "Computing" tab. Next to the "ndt_mapping" checkbox is a button labeled "[app]." Select a destination folder path and then click "PCD OUTPUT," which will export the final point cloud in PCD format.

4) **ICP Registration and Merging** – The fourth stage is using CloudCompare [17] to merge multiple PCDs into a single PCD. For small test sites, it is reasonable to follow the previous stages in order to generate a PCD from a series of consecutive PCAPs. However, the entire process operates under the assumption that the PCAPs were recorded consecutively, with no shifts in the position of the data collection vehicle. This means that the entire location would have to be scanned in one attempt, which is not feasible for larger areas. Thus, in order to generate a high-definition map for a road network, it is necessary to repeat the process several times, creating a series of PCDs that describe different sections of the entire region. These PCDs must then be merged into a single PCD before it can be used as a high-definition map.

   There are some restrictions that must first be considered when attempting to merge multiple point clouds together. First of all, the two point clouds that are being merged must share some common points, meaning that they overlap in certain places. Without these frames of reference, it would be impossible to correctly register them into a single point cloud. Additionally, merging point clouds generated using scans from two differently-positioned LiDARs interferes with the effectiveness of localization when the PCD is given to Autoware.

   CloudCompare is an open-source project devoted towards developing 3D point cloud and mesh processing software. It is especially useful because it allows users to perform complex transformations on multiple point clouds, while also providing a responsive interface. After

launching the CloudCompare visualizer, load the PCDs into the application. Select two PCDs to merge first. Manually transform one of the point clouds using the translation and rotation commands so that their shared points are roughly overlaid. Then, use the Iterative Closest Point (ICP) [18] algorithm in order to register the point clouds. Before the ICP registration is performed, input the theoretical overlap of the two point clouds (e.g. what percentage of both point clouds will be shared with one another). Merge the two registered point clouds into a single point cloud. Repeat the entire process, adding a new point cloud to the merged point cloud, until there is only a single point cloud. Save and export the fully registered point cloud as a PCD. The resulting PCD is the point cloud data component of the high-definition map.

5) **Road Logic Annotation** – The fifth stage is using VectorMapper [19] in order to generate a series of vector map files in CSV format from a PCD. VectorMapper is an online tool developed by MapTools [20] that allows users to create a vector map from point cloud data. Start by uploading the PCD on the "FILE UPLOAD" tab. Then, click on the button labeled "OPEN Vector Mapper" in order to launch the tool. Under the "File" dropdown menu, click "Open PCD file…" and select the uploaded PCD contained in "/home/autoware/Uploads." Add the necessary road logic annotations using the tools under the "Add" dropdown menu. Under the "File" dropdown menu, click "Save ADAS dir…" and save the file in "/home/autoware/Downloads." Go to the "FILE DOWNLOAD" tab and download all of the CSV files. The sum of these CSVs is the vector map component of the high-definition map.

6) **(Optional) Localization Testing** – After completing the high-definition map generation procedure, the resulting PCD map can be tested by seeing if Autoware's localization module is able to successfully locate its position on the map using only LiDAR scans of the same area. First, in the "Simulation" tab, select a ROSBAG [21] file using the "Ref ROSBAG" button. This ROSBAG should contain LiDAR scan data from the mapped area. The ROSBAG can be created from the concatenated PCAP. In the "Sensing" tab, select the applicable LiDAR model in the "LiDARs" category. After clicking the "ROSBAG" button, click "Refresh" and select "/points_raw." In "ROSBAG Record," click "Start." In terminal, run the command:

```
$ rosrun velodyne_driver velodyne_node _model:=the_model_type
_read_once:=true _pcap:"=/the/pcap/path/data.pcap"
```

When the terminal has finished reading the PCAP, click "Stop" in "ROSBAG Record." Finally, export the newly-created ROSBAG. Reference the ROSBAG in the "Simulation" tab.

Click "Play" to initialize the simulation. Once it has finished initializing and begins running, press "Pause." In the "Setup" tab, select "Velodyne" under "Localizer," activate "TF" and activate "Vehicle Model." In the "Map" tab, reference and activate the "Point Cloud" module using the PCD file. Reference and activate the "TF" module with the launch file "/Autoware/ros/src/.config/tf/tf_local.launch." In the "Sensing" tab, activate "voxel_grid_filter." In the "Computing" tab, activate "ndt_matching." Return to the "Simulation" tab and click "Play" to resume the simulation. Launch RViz and select the configuration "~/Autoware/ros/src/.config/rviz/default.rviz." Using the "2D Pose Estimate" tool, select the starting position of the vehicle path described in the ROSBAG. Repeat until the flashing LiDAR scan lines align with the topography of the PCD map. Watch as the vehicle symbol retraces the original path of the data collection vehicle performing the LiDAR scan.

VI. CONCLUSION

This research shows how autonomous vehicle researchers can easily generate high-definition maps by following the procedure detailed above. First, collect LiDAR data using a vehicle with a mounted LiDAR. Second, use Wireshark to concatenate the multiple PCAP files created by the LiDAR into a single PCAP file. Third, use Autoware to process the point cloud instances within the PCAP file using the Normal Distributions Transform algorithm in order to produce a PCD file. Fourth, the PCD file can be registered with other PCD files using the Iterative Closest Point algorithm, and then merged together within CloudCompare to produce the final PCD. Fifth, VectorMapper can be used to add road logic annotations to the PCD and generate a series of CSV vector maps. The PCD and the CSVs form the final high-definition map. This map can then be tested by performing a localization simulation in Autoware.

While the above process greatly increases the accessibility of high-definition map generation to researchers, producing an HD map using this technique still has some limitations that must be considered. As stated above, this procedure assumes that a variety of pre-conditions are correctly met, including the fact that LiDAR data must be collected in one attempt for each non-merged PCD produced. Additionally, as the size of the map grows larger, the computational power required increases dramatically. Finally, the resulting map is definitely compatible with Autoware, but is not necessarily portable to other self-driving software platforms.

Despite the limitations mentioned above, this process provides a wide variety of benefits over traditional methods of obtaining a high-definition map, such as outsourcing the map creation to a professional vendor. First, the software needed for this process, including Autoware, is open-source. This means that anyone can have access to high-definition map generation capabilities. Meanwhile, this process produces high-definition maps in the PCD and CSV file formats, both of which have gained widespread acceptance. Finally, the entire process only mandates the collection of LiDAR data, greatly reducing the barriers to performing autonomous vehicle research.

In summary, this procedure greatly reduces barriers to performing self-driving car research by extending high-definition map generation capabilities to more scientists. Therefore, it helps accelerate the creation of advanced driverless technologies and expedites the realization of an autonomous transportation future that will greatly improve human quality-of-life. Autonomous vehicles could drastically reduce traffic fatalities, alleviate urban congestion, reduce travel costs, limit harmful emissions, improve energy conservation, and increase mobility for all.

VII. FUTURE WORK

There are a wide variety of new research topics that can build upon the discoveries made over the course of this research into high-definition map generation for self-driving vehicles. While one of the main advantages of this process is its low data requirements, only needing LiDAR scans to function, it would be apt to research how to integrate other sensor inputs into this procedure. Additionally, it is important to test the quality of the high-definition map generated by this

procedure compared to a high-definition map created by a professional source. This will help identify inconsistencies and refine the process to reduce the level of error.

It would be helpful to research how to convert the Autoware high-definition map into a format used by other popular self-driving software platforms, such as Apollo. Apollo is a high-performance, flexible architecture developed by Baidu, which is designed to provide an integrated platform for the development, testing, and deployment of autonomous vehicles. Apollo uses a custom high-definition map format known as Apollo OpenDRIVE [22]. Finding a method to easily convert between Autoware's PCD/CSV maps and Apollo's OpenDRIVE maps would allow this HD map generation procedure to produce Apollo HD maps. This would greatly expand the accessibility of both self-driving platforms to clients and researchers alike.

Another major scientific question is what type of digital infrastructure will be best suited for generating high-definition maps on a global scale. At the moment, despite their fundamental importance to the operation of autonomous vehicles, high-definition maps only cover a small area of the world. In order for truly universal autonomous transportation to function, high-definition maps need to be extended globally so that self-driving cars can navigate everywhere. Solving this question will require researching the computational systems needed to effectively process that data into usable high-definition maps, as well as methods for coordinating a fleet of mapping vehicles.

VIII. REFERENCES

[1] "Autoware," Autoware, [Online]. Available: https://autoware.ai/.

[2] "ROS," Open Source Robotics Foundation, [Online]. Available: http://www.ros.org/.

[3] "Development of ICAVE2," University at Buffalo, 2016. [Online]. Available: https://icave2.cse.buffalo.edu/index.htm.

[4] "Motion Simulation Lab," University at Buffalo, 2018. [Online]. Available: https://www.buffalo.edu/shared-facilities-equip/facilities-equipment/Motion-Simulation-Lab.html.

[5] "Apollo," Baidu, 2018. [Online]. Available: http://apollo.auto/.

[6] "PUCK™ VLP-16," Velodyne LiDAR, Inc., 2018. [Online]. Available: https://velodynelidar.com/vlp-16.html.

[7] "Meet Olli," Local Motors, [Online]. Available: https://localmotors.com/meet-olli/.

[8] "Ubuntu," Canonical Ltd., 2018. [Online]. Available: https://www.ubuntu.com/.

[9] T. Chiang, "High-Definition Map Generation for Self-Driving Cars Using Autoware," 25 September 2018. [Online]. Available: https://youtu.be/T57DG7EZemU.

[10] "The PCD (Point Cloud Data) file format," PCL, [Online]. Available: http://pointclouds.org/documentation/tutorials/pcd_file_format.php.

[11] "VeloView," Kitware, [Online]. Available: https://www.paraview.org/veloview/.

[12] "ParaView," Kitware, [Online]. Available: https://www.paraview.org/.

[13] "About Wireshark," Wireshark Foundation, [Online]. Available: https://www.wireshark.org/.

[14] "PCL," PCL, [Online]. Available: http://pointclouds.org/.

[15] P. Biber, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.7059&rep=rep1&type=pdf.

[16] "rviz," Open Source Robotics Foundation, [Online]. Available: http://wiki.ros.org/rviz.

[17] "CloudCompare," CloudCompare, [Online]. Available: https://www.danielgm.net/cc/.

[18] B. L. J. Y. S. L. a. J. H. Ying He, "An Iterative Closest Points Algorithm for Registration of 3D Laser Scanner Point Clouds with Geometric Features," MDPI, 11 August 2017. [Online]. Available: www.mdpi.com/1424-8220/17/8/1862/pdf.

[19] "VectorMapper," Tier IV, Inc., 2017. [Online]. Available: https://maptools.tier4.jp/vector_mapper_description/.

[20] "MapTools," Tier IV, Inc., 2017. [Online]. Available: https://maptools.tier4.jp/.

[21] "rosbag," Open Source Robotics Foundation, 16 June 2015. [Online]. Available: http://wiki.ros.org/rosbag.

[22] "Welcome to the World of OpenDRIVE®!," VIRES Simulationstechnologie GmbH, 2018. [Online]. Available: http://www.opendrive.org/.