Rapid Procedural Generation of Real World Environments for Autonomous Vehicle Testing

Yuxiang Feng, Qiming Ye and Panagiotis Angeloudis

***Abstract*— This study proposes an automated virtual test creation pipeline that replicates real-world environments in simulation for autonomous vehicle development. By integrating publicly available tools, the pipeline accelerates AV testing with minimal human intervention. Our approach develops real-world maps for open-source simulators like CARLA and LGSVL, facilitating testing and development of most existing autonomous driving controllers, including Autoware and Apollo. This study outlines the methodology and results of our approach, highlighting its potential to significantly enhance the scalability and efficiency of autonomous vehicle testing and development.**

I. Introduction

The recent emergence and intricate nature of autonomous vehicle (AV) technology have made establishing standard testing and verification procedures challenging [1]. Current automotive testing procedures, such as the Federal Motor Vehicle Safety Standards [2] and ISO 26262 [3], focus on safety-related components and design features but do not encompass driving intelligence assessment. Given the potential catastrophic consequences of hidden errors in driving intelligence, various testing approaches, such as virtual simulation, X-in-the-loop simulation, closed-track testing, and on-road testing, have been proposed.

The advantages of various types of testing are complementary and vital to the successful deployment of AVs, despite their budget and complexity differences. Virtual testing offers scalability and cost benefits in driving intelligence development and evaluation. However, most current methods rely on manual map creation, requiring substantial human intervention and time expenditure. As suggested by the US Department of Defence, the manual creation and touch-up required to generate high-resolution geospecific terrain models for simulation takes several months [4]. Additionally, seamless integration of AV controllers like Autoware or Apollo with customized maps remains elusive. Furthermore, there is currently no mechanism to seamlessly integrate an AV controller such as Autoware or Apollo with customized maps. For instance, Autoware requires two specific file types, namely, Point Cloud Data (PCD) and vector maps, to enable its operation within any simulation environment. The PCD data facilitates vehicle localization while vector maps are vital for path planning. Both files must be appropriately aligned to enable Autoware to function with customized maps.

The authors are with the Transport Systems & Logistics Laboratory, Department of Civil and Environmental Engineering, Imperial College London, UK. All correspondence should be addressed to y.feng19@imperial.ac.uk

The available literature on the topic is relatively scarce, with only two studies being identified. De Miguel et al. [5] presented a process for converting OpenStreetMaps to CARLA map. However, this method only incorporates road layout and does not include buildings or other terrain objects, which limits its applicability to LIDAR-based external AV control stack. Another study by Mondal et al. [6] also described a similar pipeline, which relies on Blender for simulation, but did not generate files required by external AV control stack.

The lack of a rapid procedural generation pipeline impedes the creation of virtual testing scenarios and consequently limits scalability and transferability from simulation to reality. To address these challenges, this paper proposes a novel pipeline that automatically generates customized maps and associated Autoware files from real-world data sources. The process aims to increase the diversity of available testing scenarios and accelerate edge-case replication from on-road driving. This approach is essential for virtual AV verification and can facilitate testing and optimization of AV control stacks. The study focuses on two popular open-source virtual simulators in AV research, namely CARLA [7] and LGSVL[8].

II. Proposed map development pipeline

A summary of the proposed pipeline is presented in Fig. 1. As the first step towards the creation of customized maps from real-world locations, a unified data source that contains comprehensive information and specifications of the environment is required. Considering its open-source nature and the availability of detailed information, OpenStreetMap (OSM) [9] was chosen as the data source for this study. The resulting OSM data consists of three primary components: node, way, and relation. Additionally, the data includes supplementary information regarding elements within the area, such as coordinates, element types, speed limit, building height, and others.

Afterwards, in order to create a map model based on the downloaded source data, various options are available. Mondal et al. [6] utilized blender-osm to construct road and building models in Blender, while commercialized map generation tools (Mapbox [10], Citygen3D [11], AccuCities [12]) could be employed with minor adaptations. The proposed pipeline can accommodate either of these options provided an original OSM file is available, and the resulting 3D map model can be exported in FBX format. In addition to the 3D map model, a road specification file is also required to define the vehicle routes while creating the map. As

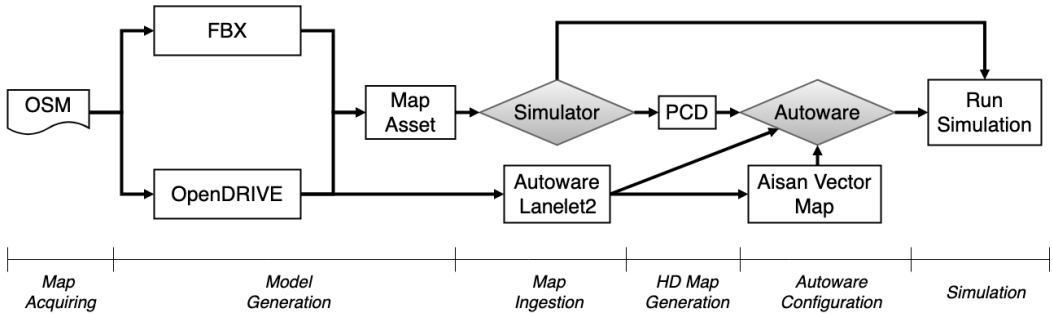

Fig. 1. Workflow of the map customisation pipeline.

both simulators support roads defined with OpenDRIVE, an automatic conversion script was developed to generate the OpenDRIVE file from the corresponding OpenStreetMap file.

With the 3D map model and OpenDRIVE file obtained, the next step is to import these files into the simulators. Both simulators have their own procedure for adding new maps. For CARLA, a json file needs to be created to specify the name of the map, the directory of those files, etc. For LGSVL, both files can be imported directly into the scene. An example of the map in LGSVL is shown in Fig. 2.

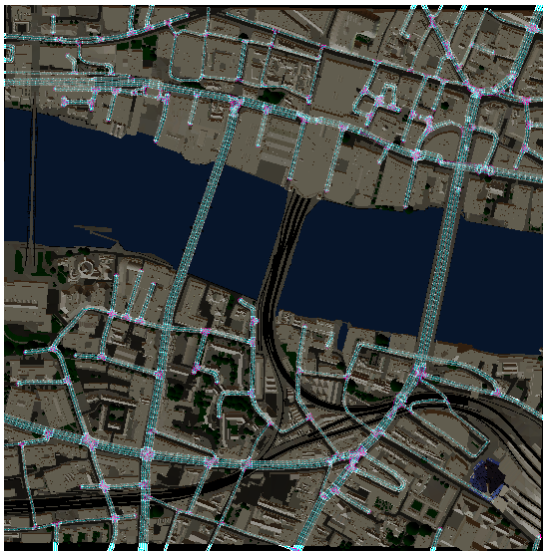

Fig. 2. Map ingested in LGSVL.

After the customised maps are created and imported in both simulators, the next step is to prepare them for testing AV control stack (Autoware, Apollo, etc.). Since the files required by most existing AV control stack are similar and interconvertible, this study focuses on the testing of Autoware. As outlined in the previous section, two types of files (PCD and vector map) are both needed to run Autoware on customised maps. To facilitate the recording of PCD, an automatic recording function was developed in CALRA. A simulated AV with LIDAR sensor is spawned at different locations in the map (with the spawn points derived from the OpenDRIVE file), and automatically navigates through the map with the CARLA's Autopilot mode while recording the PCD data of its surrounding environment. This process will run iteratively and will stop automatically once the entire map is covered. These PCD files can then be concatenated to form the PCD of the customised map, as shown in Fig. 6. Meanwhile, the required vector map (Autoware LaneLet2 and Aisan vector map) can be generated from the corresponding OpenDRIVE file. These files can then be specified in Autoware to facilitate AV testing on the customised maps.

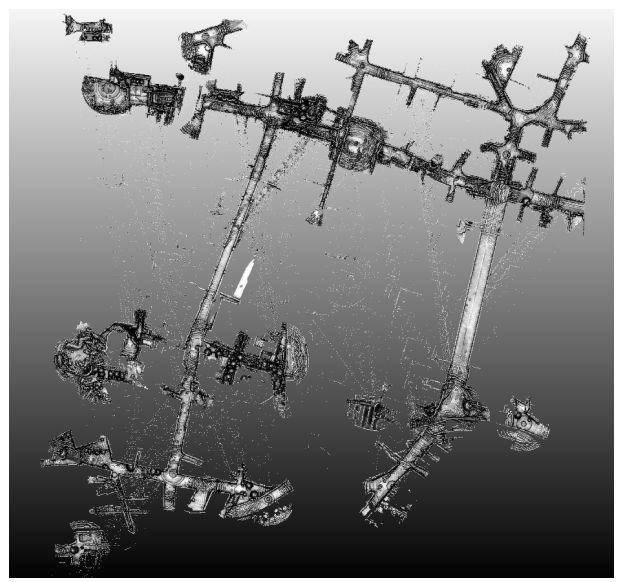

Fig. 3. PCD of the customised map.

III. Conclusions

In this study, we developed a pipeline for automated procedural generation of customized maps from real-world environments. This pipeline minimizes human intervention and creates simulation-ready maps for CARLA and LGSVL, facilitating the testing of AV control stacks. Existing AV controllers like Autoware and Apollo can be tested in customized maps with LIDAR-based localization and camera-based perception systems functioning seamlessly. Furthermore, traffic simulation software such as SUMO can be used to create realistic traffic scenarios. Although focused on CARLA, LGSVL, and Autoware, the pipeline can be extended to other simulators and AV control stacks.

References

[1] S. Feng, X. T. Yan, H. W. Sun, Y. H. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Communications*, vol. 12, 2021.

[2] National Highway Traffic Safety Administration, "Federal motor vehicle safety standards," Washington, D.C., Tech. Rep., 1999.

[3] ISO Standard, "Iso: Iso 26262 road vehicles– functional safety," Geneva, Tech. Rep., 2011.

[4] SBIR, "Rapid Terrain/Map Generation for Robotic and Autonomous Vehicle Simulations," https://www.sbir.gov/node/1696677, 2020, last accessed 21 July 2021.

[5] M. Á. de Miguel, F. M. Moreno, P. Marín-Plaza, A. Al-Kaff, M. Palos, D. Martín, R. Encinar-Martín, and F. García, "A research platform for autonomous vehicles technologies research in the insurance sector," *Applied Sciences (Switzerland)*, vol. 10, no. 16, 2020.

[6] A. Mondal, "Real2sim : Automatic generation of open street map towns for autonomous driving benchmarks," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," https://arxiv.org/abs/1711.03938, 2017.

[8] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "Lgsvl simulator: A high fidelity simulator for autonomous driving," https://arxiv.org/abs/2005.03778, 2020.

[9] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," https://www.openstreetmap.org, 2017, last accessed 21 July 2021.

[10] Mapbox, "Maps, geocoding, and navigation APIs and SDKs — Mapbox," https://www.mapbox.com/, 2021, last accessed 21 July 2021.

[11] CityGen3D, "Citygen3d — procedural city generation from map data," https://forum.unity.com/threads/released-citygen3d-procedural-city-generation-from-map-data.514677, 2020, last accessed 22 July 2021.

[12] AccuCities, "3d model of london and 3d city models," https://www.accucities.com, 2021, last accessed 21 July 2021.

Appendix

A. Example location in OpenStreetMap

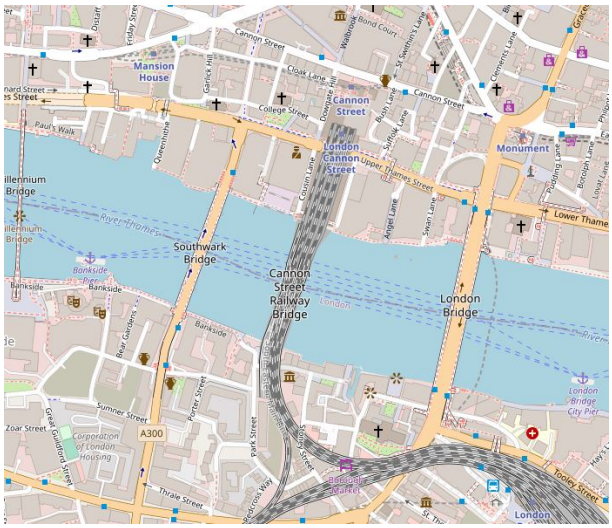

Fig. 4. Example from OpenStreetMap.

B. FBX model example from AccuCities

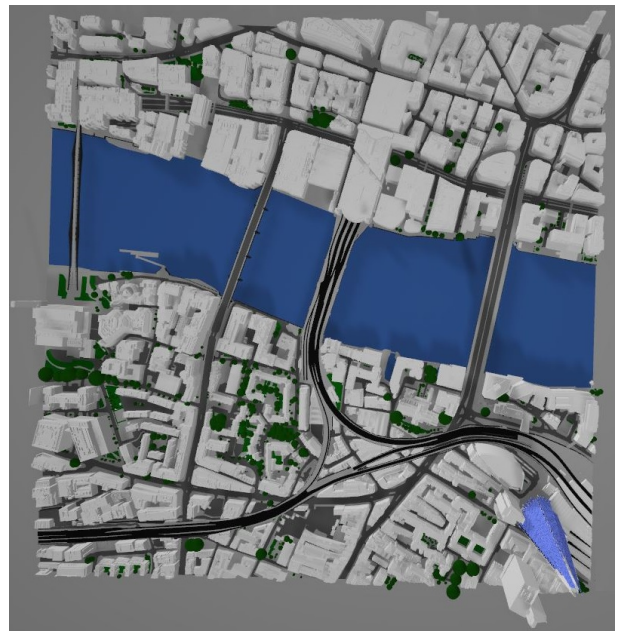

Fig. 5. FBX model of the map.

C. Example map in CARLA

Fig. 6. Map ingested in CARLA.