Development of an Information Extraction System for Mobile LiDAR Survey Data using Free and Open-Source Technologies

Richard Watson[1], Peter Ryan[1] and Leigh Carnall[2]

1. Ryan Watson Consulting Pty Ltd
2. Civiltech Solutions Pty Ltd

Abstract

A WebGIS Information Extraction System was developed to retrieve and visualize Mobile LiDAR Survey data. It uses Python, JavaScript and HTML, together with the PostgreSQL/PostGIS, FastAPI and Leaflet packages which are all free and open-source technologies. The system aims to have user-friendly interfaces and to be fully open-source development. The system is not yet complete; however some preliminary results are given here.

1 Introduction

Rapid advancements in technology have improved 3D visualization techniques in Geographic Information Systems (GIS). Open-source solutions can provide the functionality to store, serve, and visualise large quantities of GIS data in multiple formats [1]. This paper describes a prototype WebGIS application for visualizing Mobile LiDAR GIS data across various Levels of Detail (LoDs).

2 Mobile LiDAR Survey Technologies

Mobile LiDAR is a powerful geospatial technology for surveying complex urban landscapes including roads and roadside infrastructure. It provides accurate 3D data by integrating laser scanning, Inertial Measurement Unit (IMU), and Global Navigation Satellite System (GNSS) technologies to produce detailed point clouds. A schematic truck-mounted Mobile LiDAR system is shown in Figure 1.

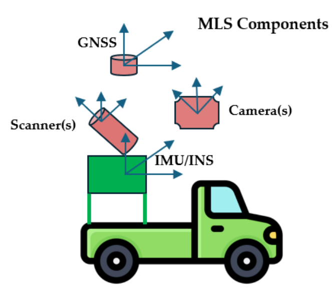

Figure 1: Mobile LiDAR system (adapted from [2])

Mobile LiDAR Surveys (MLS) capture X, Y, Z coordinates of laser light reflecting objects along with attributes like intensity and RGB colour values. They also capture imagery that can be fused with the point cloud data to improve object classification. To identify and semantically classify objects such as roadside features requires the use of Artificial Intelligence (AI) software which incorporates complex algorithms which may be tailored to particular types of object. Thus detecting the road surface may require a different algorithm from pole-like objects such as street signs.

This study is based on survey data captured by Cyvl[1] LiDAR and 360-degree optical imagery sensors mounted on vehicles which travel on targeted roads at the same speed as other traffic. The Cyvl system also contains a suite of Artificial Intelligence (AI) software which is mainly used for road inventory recording, defect detection and road safety management. Road defects include road pavement damage and potholes. Road safety issues include street trees encroaching on the road, damaged signage, lighting fixtures and lane markings and obstacles to driving. The road infrastructure data relevant to this project are indicated in the diagram below.

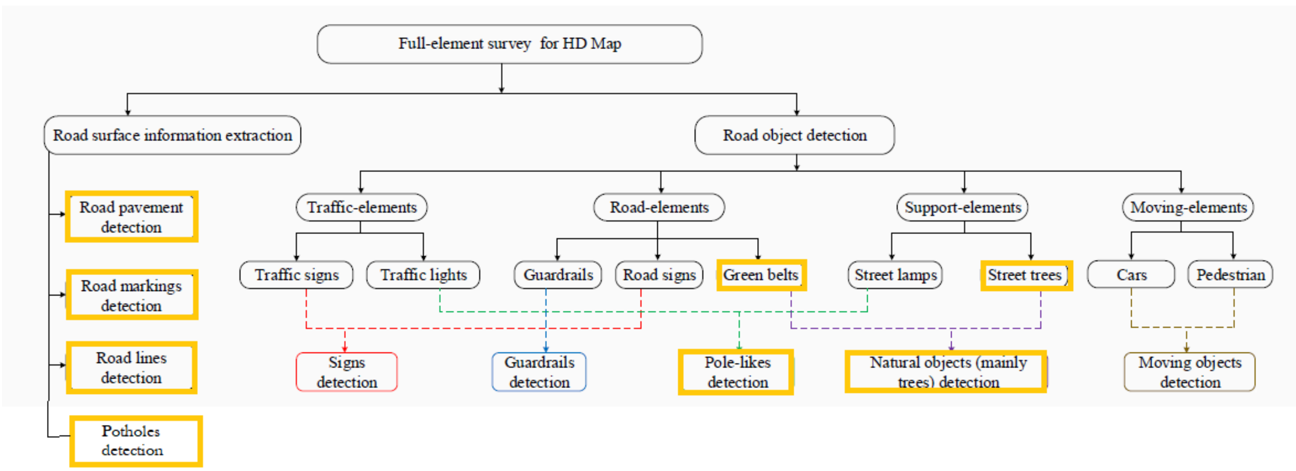

Figure 2: Taxonomy of elements for road surface and object identification (adapted from [3]). Features relevant to this project are indicated by orange boxes.

The Cyvl system identifies and semantically classifies roadside features captured by its optical imagery sensors and matches these to the point cloud data collected by its LiDAR sensor. This is a more accurate method of identifying roadside features than methods which rely on LiDAR data alone, see e.g. [4].

An additional type of infrastructure feature not shown above is powerlines, both roadside and cross country. The Cyvl system can identify powerlines as well as vegetation which may interfere with their operation., and the project may involve development of specialized algorithms to assess the risk and advise on vegetation removal action.

[1] https://www.cyvl.com/

3 Information Extraction System Overview

Roadside objects are segmented using the Cyvl AI system. Each object is stored in an ontological format, reducing data size for efficient querying. The integration process results in a semantically rich master geo-database for various roadside feature classes.

The information extraction system described here, developed by Ryan Watson Consulting (RWC), supplements the Cyvl AI system by making the data more accessible and allowing for various types of queries, including spatial location-based, attribute-based, and aggregate queries. A system architecture diagram (simplified) is depicted in Figure 3 below. It will be an ensemble of:

(i) data pre-processing using Python packages including GeoPandas, GDAL, Rasterio and PDAL.

(ii) data integration, schema development and 3D database development based on PostgreSQL/PosrGIS [4].

(iii) asynchronous application programming interface (API) based on FastAPI [5], serving both data and auto-generated documentation [6].

(iv) a front-end meeting client applications including maps, charts, tables and reports.

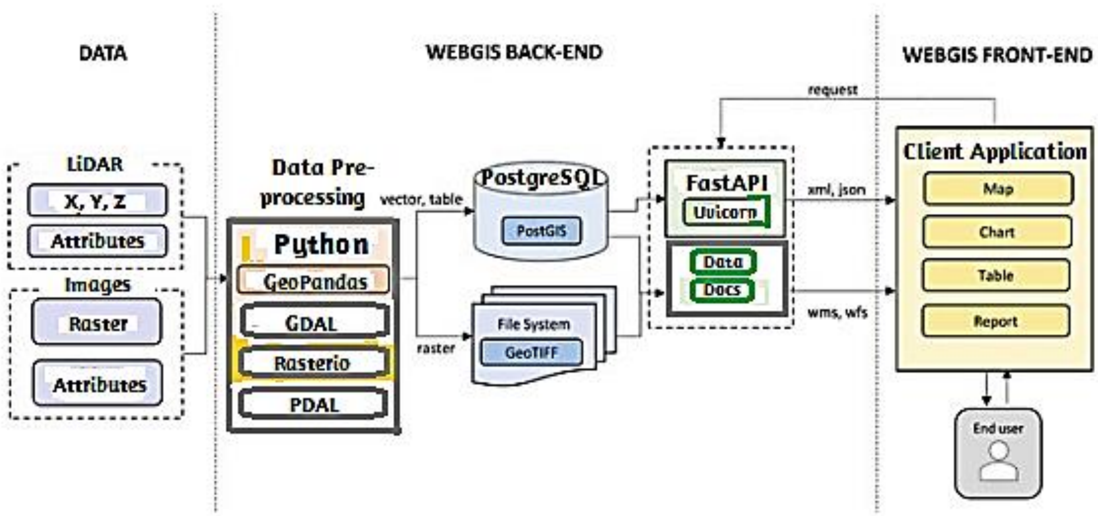

Figure 3: System Architecture

4 Preliminary Results

4.1 Shape file served by FastAPI

FastAPI [5] is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints. It offers robust and automatic API documentation generation as a core feature. This functionality is built upon the OpenAPI specification and provides interactive documentation interfaces, primarily Swagger UI and ReDoc [6]. The methodology described in ref [7] was followed, however a shape file of the Melbourne metropolitan area (using EPSG:4326) was used instead of a European one. This was served as JSON by FastAPI and rendered by the Leaflet package overlaid on an Open Street Map (OSM) [8] base as shown in Figure 4.

Figure 4: Leaflet screenshot of Melbourne metro area shape files served with FastAPI with OSM background

4.2 GeoTIFF tiles served by GeoServer

Point cloud data (PCD) provided by the Cyvl system was transformed into GeoTIFF tiles (EPSG: 28355) by RWC analysts. An image mosaic using 16 GeoTIFF tiles from Queensland's Gold Coast and an OSM base was produced using GeoServer [9] and is depicted in Figure 5. The map added grayscale Styled Layer Descriptor (SLD) for better visibility

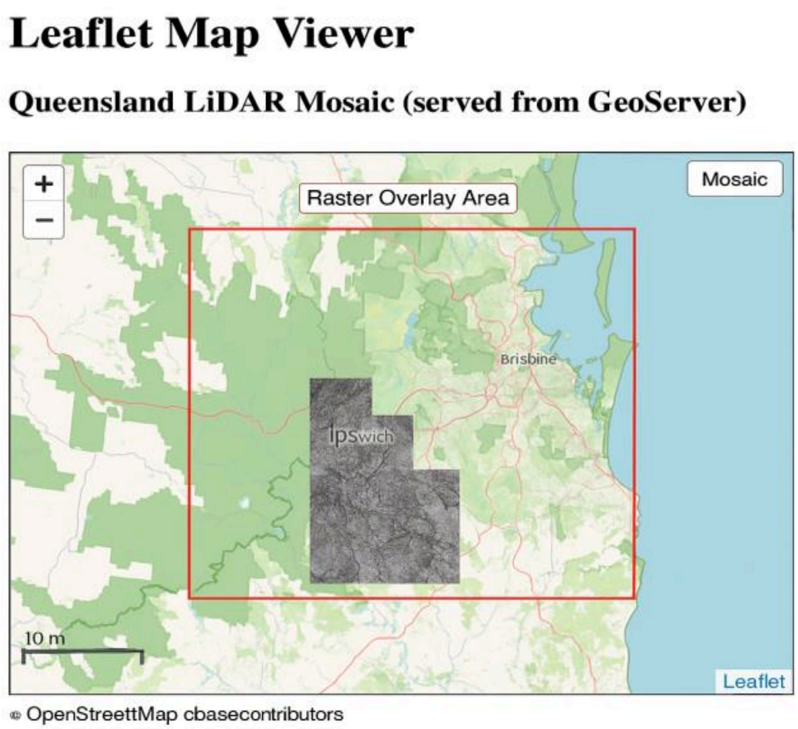

Figure 5: Leaflet screenshot of 16 GeoTIFF tiles as a mosaic

5 Conclusions

The developed WebGIS applications effectively extract and visualize Mobile LiDAR survey data. This will help road authorities make more effective use of the data being collected. Future applications may include complex queries such as assigning Pavement Condition Index (PCI) and road safety ratings such as the International Road Assessment Program (iRAP) star rating.

6 References

1. Pirotti, F. (2019). Open software and standards in the realm of laser scanning technology. *Open Geospatial Data, Software and Standards* **4** (1) 2019/11/21 14 DOI: 10.1186/s40965-019-0073-z
2. National Cooperative Highway Research Program (US) (2025). Chapter 9: Typical components of mobile LIDAR systems. In: Transportation Research Board of the National Academies (ed.) *Guidelines for the Use of Mobile LIDAR in Transportation Applications*. Available from: https://learnmobilelidar.com/guidance-document/chapter-9-background/.
3. Luo, Z., Gao, L., Xiang, H. and Li, J. (2023). Road object detection for HD map: Full-element survey, analysis and perspectives. *ISPRS Journal of Photogrammetry and Remote Sensing* **197** 2023/03/01/ 122-144 DOI: https://doi.org/10.1016/j.isprsjprs.2023.01.009
4. Abro, G.-E. M., Zahid, F., Rajput, S., Azhar Ali, S. S. and Aromoye, I. A. (2025). Challenges and Innovations in 3D Object Recognition: The Integration of LiDAR and Camera Sensors for Autonomous Applications. *Transportation Research Procedia* **84** 2025/01/01/ 618-624 DOI: https://doi.org/10.1016/j.trpro.2025.03.116
5. fastAPI. [Accessed: 10 September 2025]; Available from: https://fastapi.tiangolo.com/.
6. Metadata and Docs URLs. [Accessed: 10 September 2025]; Available from: https://fastapi.tiangolo.com/tutorial/metadata/.

7. Machairas, I. (2024). Combining FastAPI, PostgreSQL, and Leaflet — GIS Tutorial. Available from: https://levelup.gitconnected.com/combining-fastapi-postgresql-and-leaflet-gis-tutorial-597e6a8ed028.
8. OpenStreetMap. [Accessed: 2 Sep 2025]; Available from: https://www.openstreetmap.org/.
9. GeoServer. [Accessed: 2 Sep 2025]; Available from: https://geoserver.org/.