

University of British Columbia

Social Ecological Economic Development Studies (SEEDS) Sustainability Program

Student Research Report

Tree Ecosystem Services Analysis Platform

Prepared by: Shaylin Cattell, Jonathan Lew, Mikayla Neufeld, Caelin Preston, Yiyi Yan

Prepared for:

Course Code: CPEN 491

University of British Columbia

Date: 29 April 2021

Disclaimer: "UBC SEEDS Sustainability Program provides students with the opportunity to share the findings of their studies, as well as their opinions, conclusions and recommendations with the UBC community. The reader should bear in mind that this is a student research project and is not an official document of UBC. Furthermore, readers should bear in mind that these reports may not reflect the current status of activities at UBC. We urge you to contact the research persons mentioned in a report or the SEEDS Sustainability Program representative about the current status of the subject matter of a report".





UBC SEEDS Report: TREESAP

Tree Ecosystem Services Analysis Platform

Capstone Team 117

Shaylin Cattell, Jonathan Lew, Mikayla Neufeld, Caelin Preston, Yiyi Yan

April 29, 2021

Table of Contents

Executive Summary	3
Appendix 1: Deliverables	5
Appendix 1.1: Database	6
Appendix 1.2: Desktop Application	6
Appendix 1.3: Data Processing Pipelines	7
Appendix 1.4: Data Acquisition Methodologies	8
Appendix 1.5: Videos	8

Executive Summary

As UBC Electrical and Computer Engineering Capstone Team 117, we worked in collaboration with UBC Campus and Community Planning and UBC SEEDS to develop a project to help support sustainable development policies and goals at UBC. Our project, the Tree Ecosystem Services Analysis Platform (TREESAP), is a tool for analysing and visualizing ecosystem services on the UBC Vancouver campus. We have built a processing pipeline that extracts locations of tree cover on campus from aerial scans, which are in LiDAR and orthophoto formats. Tree clusters are identified in orthophoto by using colour and standard deviation to segment and separate trees from other areas. LiDAR processing consists of a 3D Density-Based Spatial Clustering (DBSCAN) clustering algorithm to identify and separate tree clusters from other points in the scan. These clusters are then smoothed using alpha shape to obtain the optimal outer contour for the cluster. We also provide a desktop app to visualize this data, allowing users to generate reports about ecosystem services, in particular, the carbon sequestration in tonnes/year and the avoided rainwater runoff in litres/year that trees provide. Additionally, shading and cooling information can be gathered by using the interface to calculate the distance between buildings and trees. This app can be used to plan future developments by seeing the potential effects of adding and removing trees. TREESAP provides baseline data for ecosystem services on campus, helping urban planners develop quantitative arboreal sustainability goals. The deliverables of this project are explained in more detail in *Appendix 1*.

Given the limited timeframe of our project, we designed the system with future extensions in mind. We've come up with several ideas that future teams could implement:

- Modify or create a new data processing pipeline to use new data sources or new approaches (e.g. machine learning) to provide more or better data. The input to the desktop application is polygons of GeoJson format, so any processing technique that produces valid GeoJson can make use of the desktop application

- Add economic value of the ecosystem services to the desktop application. For example, trees near a cliff may be much more valuable for avoided runoff than those in the middle of campus
- Add more ecosystem services to the desktop application. iTree Canopy¹ provides many more ecosystem benefits that depend only on tree cover area, so adding these would work within the architecture of our desktop application.
- Update and scale desktop app so it can be publicly accessible as a tool for public engagement. This may involve some service fees for Google Maps API, for example, on top of usability improvements.

¹ <https://canopy.itreetools.org/benefits>

Appendix 1: Deliverables

TREESAP is composed of the following deliverables: a database containing processed data from the data processing pipeline, along with other files necessary for the desktop application; source code for the data processing pipeline and the desktop application; and a document outlining the data sources used or considered and the methodologies for obtaining them. Specifically, the items we delivered are:

1. A link to Github repo containing database of GeoJSON files of tree cover data
 - a. One file for each data source processed (2018 LiDAR, 2014-2020 Orthophoto)
 - b. One main database file that is used by the visualization application to obtain tree cover data
 - c. One file containing ecosystem services parameters
2. Link to Github repo containing all source code for the project, with each component in a separate folder
 - a. Source code for the desktop application
 - b. Executable file to run the desktop application that includes all packages, libraries, and dependencies
 - c. Source code for LiDAR pipeline
 - i. Includes python scripts that analyze the LiDAR scans
 - d. Source code for Orthophoto pipeline
 - i. Includes python scripts that analyze the orthophotos
3. Data acquisition methodologies document

More information about each deliverable can be found in the following sections.

Appendix 1.1: Database

Our database is available as a public Github repository at <https://github.com/Capstone-TREESAP/TREESAP-Database>.

Appendix 1.2: Desktop Application

The desktop application source code is available in a public Github repository at https://github.com/Capstone-TREESAP/TREESAP/tree/main/desktop_application. The README on this page contains a guide for setting up the application as an executable or in development mode.

Additionally, there is documentation provided for the application:

- An extensive [desktop app user guide](#)
- A [manual test plan](#) to ensure the app is working correctly

The desktop application was designed to meet the following requirements:

1. Maps tree cover as a set of polygons that outline the boundaries of tree clusters, which are a group of trees that are directly connected by their canopy or are separated by a negligible distance. Tree cover polygons are displayed as polygons overlaid on a basemap provided by the Google Maps API².
2. Users have the ability to select and hide visualization layers for processed data from previous years and for predefined areas of interest on campus, such as specific neighbourhoods.
3. Users have the ability to display more than one set of tree cover polygons at a time.
4. Users have the ability to edit the points that make up a tree cover polygon.
5. Users have the ability to delete or add new polygons to a tree cover polygon layer.

² <https://developers.google.com/maps>

6. Displays ecosystem services for each tree cover polygon as defined in *Appendix B: Definition of Ecosystem Services*, in an info window when a polygon is selected.
7. Users have the ability to select an arbitrary area on the map to view information about tree clusters within that area, including area of tree cover, carbon sequestration values, and avoided rainwater runoff values.
8. Users have the ability to display building polygons overlaid on the basemap.
9. Users have the ability to select a tree cover polygon and a building polygon, and calculate the distance between the two polygons.
10. Users have the ability to generate a report about ecosystem services contained within a user-defined intersection polygon or a predefined area of interest polygon.
11. Users have the ability to test different values for the ecosystem service parameters and see the changes to ecosystem service values per tree cluster on the map.

Appendix 1.3: Data Processing Pipelines

The GUI to run both pipelines is described in this markdown file:

<https://github.com/Capstone-TREESAP/TREESAP/blob/main/lidar/resource/GUI.md>

The data processing pipeline is designed to meet the following requirements:

1. The dataset should provide tree cover data for the area within the legal boundary of the UBC Vancouver campus.
2. The output data of the data processing pipeline is the location of tree cover in the form of a set of polygons stored in a GeoJSON³ file. Each vertex in each polygon is defined with an absolute location as longitude and latitude values.

³ <https://geojson.org/>

3. Users can generate a new set of tree cover polygons in a GeoJSON file by providing a new raw dataset in the form of an orthophoto or LiDAR scan of campus.
4. Users have the ability to manually tune data processing parameters in the data processing pipeline GUI.

Appendix 1.4: Data Acquisition Methodologies

Documentation is available for the data acquisition methodologies used in TREESAP. This includes data sources that we used and how we obtained them, along with data sources considered but not used. The full document is available [here](#).

Appendix 1.5: Videos

We have recorded two videos for our project:

- A two-minute high level summary video, available at https://youtu.be/7CRjPUR_6X4.
- A video demonstrating the use of our desktop app, available [here](#).