The Journal of Open Source Software

PyTLiDAR: A Python Package for Tree QSM Modeling from Terrestrial LiDAR Data

John Hagood[1,2¶], **Fan Yang**[1,2¶], **Jeffery B. Cannon**[3¶], **Shruti Motiwale**[1,4], **and James Stroud**[5¶]

1 Human Augmented Analytics Groups (HAAG), Georgia Institute of Technology, United States **2** School of Computer Science, Georgia Institute of Technology, United States **3** The Jones Center at Ichauway, United States **4** University of Texas at Austin, United States **5** School of Biological Sciences, Georgia Institute of Technology, United States ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

PyTLiDAR is an open-source Python package that reconstructs 3D tree Quantitative Structure Models (QSM) from Terresrial LiDAR Scan (TLS) data, providing a user-friendly tool improving and expanding based on the MATLAB-based TreeQSM method (P. Raumonen et al., 2013). PyTLiDAR provides an accessible, extensible, and GUI-driven workflow for researchers and practitioners in forestry, ecology, and 3D vegetation modeling. The package also integrates interactive visualization tools for inspecting model quality and derived tree metrics.

Key features of PyTLiDAR include:

-A full reimplementation of TreeQSM's core logic in Python

-A user-friendly GUI built with PyQt6

-Automated and manual configuration of model generation parameters, including patch diameter ranges

-Support for interactive 3D visualization of tree models and parameter tuning

-Batch or single-file processing

Statement of Need

Terrestrial Laser Scanning (TLS) is an active remote sensing technology which uses infrared laser pulses to collect millions of points on the surface of objects, preserving spatial information and providing unprecedented detail on structural information. The technology is rapidly being adopted for diverse uses in forestry and ecology, as it is useful for estimating forest structure (Donager et al., 2021), Above Ground Biomass (AGB) (Atkins et al., 2025), gap fraction and forest fuels (Loudermilk et al., 2023), crown shape (Zhu et al., 2020), disturbance patterns (Cannon et al., 2024), tree competition (Metz et al., 2013), physiology (Hakala et al., 2015),and other ecological properties. To fully realize the potential of TLS for these applications, accurate and efficient reconstruction of QSMs from TLS point cloud data is essential (Hackenberg et al., 2015).

The use of QSM software on point cloud data permits estimation of detailed components of branch architecture such as branch diameter, volume, and distribution along the trunk (Lau et al., 2018), providing detailed information for fine-scale estimates of AGB, canopy architecture, and more. TreeQSM is a software that has been widely used in forestry and ecology for modeling tree structures from TLS point clouds (Terryn et al., 2020). Comparing to other similar softwares, TreeQSM stands out for speed, reliability, and ease of use, while Computree offers broad functionality but suffers from a large installation size and less intuitive interface. AdQSM is extremely fast and simple but lacks advanced features and source code access. aRchi provides various functions but is slow, not sufficiently documented, and harder to set up. 3dForest has a promising GUI but is currently unstable, crashing when loading data. Thus, we would like to focus on porting and improving TreeQSM. Its reliance on MATLAB makes it less accessible for users without a commercial license or familiarity with the MATLAB environment. Furthermore, the lack of a graphical interface makes the tool less user-friendly and its parameter tuning less efficient.

PyTLiDAR addresses these issues by providing a native Python implementation of TreeQSM's core algorithms, wrapped in a streamlined graphical interface that allows researchers to visualize and evaluate their models. It promotes reproducible and exploratory research by offering transparent parameter control, open-source licensing, and seamless integration into Python-based analysis workflows. This work lowers the barrier for adoption of QSM modeling by removing the MATLAB dependency, enhancing accessibility for the broader open-source geospatial and ecological modeling community.

Method

TreeQSM models individual trees from terrestrial LiDAR scans by covering the input point cloud with small, connected surface patches. These patches form the building blocks for reconstructing the tree's global shape. The algorithm first identifies these surface patches using local geometric properties, then establishes neighbor relationships between adjacent patches. Based on neighbor relationships of the surface patches, the point cloud is segmented into individual branches, with parent-children relationships of branches recorded. Then each branch is approximated as a collection of connected cylinders of varying radius, length, and orientation. This cylinder-based representation offers a simple yet effective regularization of the complex tree structure, supporting downstream analyses such as stem volume estimation or structural trait extraction (Pasi Raumonen et al., 2013) (Markku et al., 2015).

Software Architecture

PyTLiDAR is organized into several key modules: core QSM algorithms (treeqsm.py), batch processing utilities (treeqsm_batch.py), GUI components built with PyQt6 (Python bindings for the Qt 6 framework), and visualization tools using Plotly. The software follows a modular design that allows researchers to either use the complete GUI application or integrate individual components into their own Python workflows.

Software Description

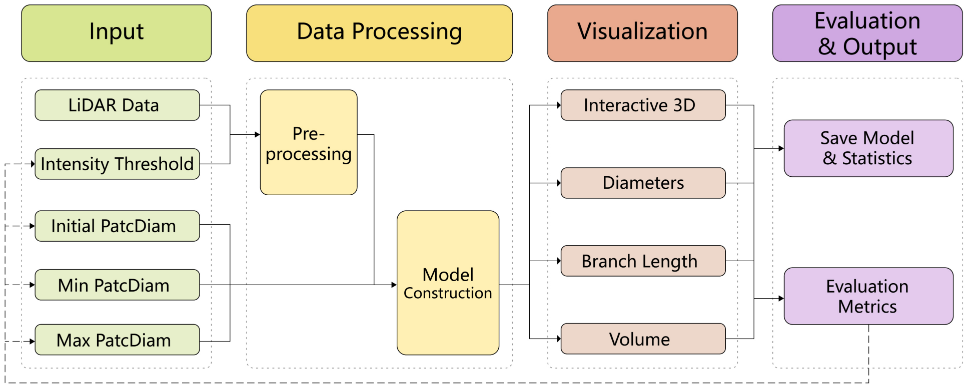

Figure 1: Software flowchart.

PyTLiDAR takes in user defined or generated parameter configuration, processes the LiDAR data and constructs QSM, then generates visualizations and analysis for the results (Figure 1). Upon launching the application, users can input or automatically generate values for key modeling parameters, including the minimum, and maximum patch diameters within a user-defined parameter space. Also, an intensity threshold can be set to filter the point cloud data, helping to remove LiDAR returns due to noise or vegetation prior to modeling (Figure 2).

Users may choose between batch processing of an entire directory of point cloud files or processing a single file. The GUI also includes options for displaying only the optimal model, based on selectable performance metrics such as 'all_mean_dis' (mean distance between point cloud and reconstructed model surface) selectable from a dropdown menu.

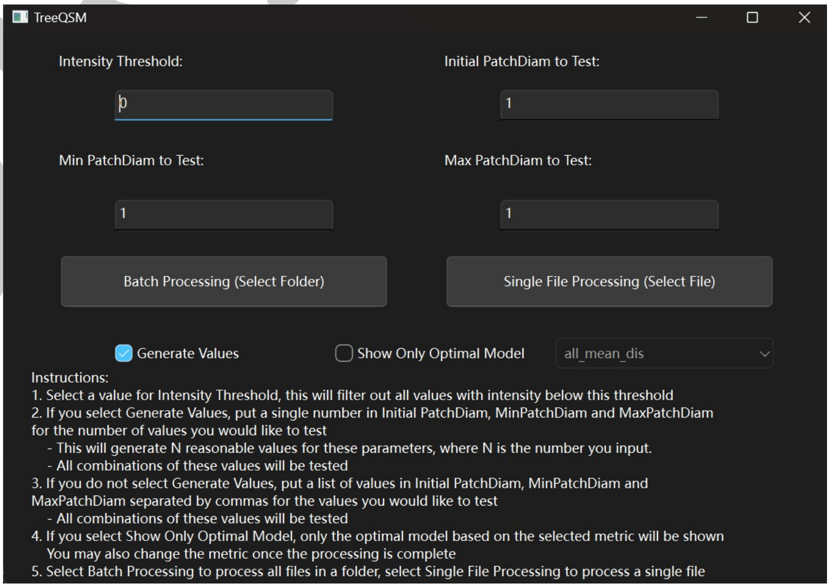

Figure 2: Software interface for user input and data selection.

After parameter and file selection, the software opens a new interface displaying data processing progress. Once the QSM reconstruction process is complete, PyTLiDAR provides interactive

3D visualization of the generated QSM using plotly (Figure 3). Users can inspect the structural fidelity of the reconstructed model, including trunk and branch geometry, and compare different parameter configurations for best fit. This combination of visual feedback and customizable processing offers an efficient path toward accurate and transparent tree structure analysis. If running in batch mode, users may also set the number of parallel cores to utilize to run simultaneous processes.

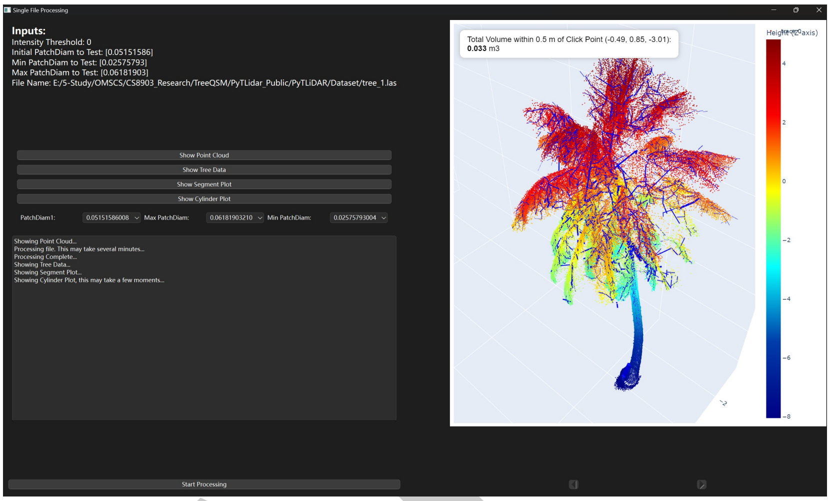

Figure 3: Software interface for processing and interactive visualization.

Users can also review the morphological summeries of the QSM, including distribution of branch diameters, branch volume, surface area, and length with regard to diameter or order from stem, as with the original TreeQSM implementation (Figure 4).

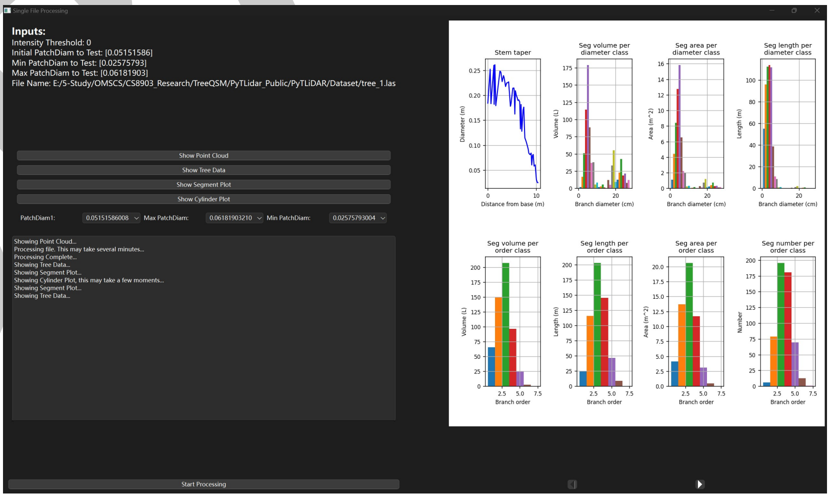

Figure 4: Tree QSM data display

Both treeqsm.py and teeqsm_batch.py may be run directly from the command line, detailed instructions are included in the documentation. This allows users to integrate the same functionality provided in the GUI into their own scripts with ease, whether those scripts are in python or not. Python users can install and use the package to get the full functionality by importing treeqsm.

[#add_benchmark_comparison]

Availability and Installation

PyTLiDAR is available at this GitHub repository. The package requires Python 3.8+ and a few key dependencies listed in the requirements. Installation instructions and example datasets are provided in the repository documentation.

Future Additions

While the initial release is focused on porting only TreeQSM, several future additions to PyTLiDAR are planned.

The first planned enhancement is to provide a novel pipeline for analyzing LiDAR scans of entire forest ecosystems to quantify vegetation structure at particular locations. This would allow users to load a series of LiDAR scan tiles and GPS observations of fauna and directly measure the environments, providing greater insights on components of habitat structural complexity.

Other planned enhancements include functions provided to users for processing LiDAR point clouds, including but not limited to both established and novel methods to perform Ground Filtering, Tree Segmentation and Leaf/Wood separation. The intended goal for this package is to provide a single source for any user processing terrestrial LiDAR to perform every step of their analysis.

Acknowledgements

We acknowledge contributions from Amir Hossein Alikhah Mishamandani during the development of the package. This work also received high level guidance from Breanna Shi, Dori P., and thanks to other staffs from the Human-Augmented Analytics Group to make this happen.

References

Atkins, D. H., Blackburn, R. C., Laughlin, D. C., Moore, M. M., & Sánchez Meador, A. J. (2025). Handheld lidar sensors can accurately measure aboveground biomass. *Ecosphere*, *16*(6), e70232. https://doi.org/https://doi.org/10.1002/ecs2.70232

Cannon, J., Zampieri, N. E., Whelan, A. W., Shearman, T. M., & Varner, J. M. (2024). Terrestrial lidar scanning reduces subjectivity in measurements of tree fire injury. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5056546

Donager, J. J., Sánchez Meador, A. J., & Blackburn, R. C. (2021). Adjudicating perspectives on forest structure: How do airborne, terrestrial, and mobile lidar-derived estimates compare? *Remote Sensing*, *13*(12). https://doi.org/10.3390/rs13122297

Hackenberg, J., Spiecker, H., Calders, K., Disney, M., & Raumonen, P. (2015). SimpleTree —an efficient open source tool to build tree models from TLS clouds. *Forests*, *6*(11), 4245–4294. https://doi.org/10.3390/f6114245

Hakala, T., Nevalainen, O., Kaasalainen, S., & Mäkipää, R. (2015). Technical note: Multispectral lidar time series of pine canopy chlorophyll content. *Biogeosciences*, *12*(5), 1629–1634. https://doi.org/10.5194/bg-12-1629-2015

Lau, A., Bentley, L. P., Martius, C., Shenkin, A., Salgado-Labouriau, M., Jackson, T., Malhi, Y., Raumonen, P., Herault, B., Baraloto, C., Phillips, O. L., Gloor, E., Asner, G. P., & Enquist, B. J. (2018). Quantifying branch architecture of tropical trees using terrestrial LiDAR and 3D modelling. *Trees*, *32*(4), 1219–1231. https://doi.org/10.1007/s00468-018-1704-1

Loudermilk, E. L., Pokswinski, S., Hawley, C. M., Maxwell, A., Gallagher, M. R., Skowronski, N. S., Hudak, A. T., Hoffman, C., & Hiers, J. K. (2023). Terrestrial laser scan metrics predict surface vegetation biomass and consumption in a frequently burned southeastern u.s. ecosystem. *Fire*, *6*(4). https://doi.org/10.3390/fire6040151

Markku, Å., Raumonen, P., Kaasalainen, M., & Casella, E. (2015). Analysis of geometric primitives in quantitative structure models of tree stems. *Remote Sensing*, *7*(4), 4581–4603. https://doi.org/10.3390/rs70404581

Metz, J., Seidel, D., Schall, P., Scheffer, D., Schulze, E.-D., & Ammer, C. (2013). Crown modeling by terrestrial laser scanning as an approach to assess the effect of aboveground intra- and interspecific competition on tree growth. *Forest Ecology and Management*, *310*, 275–288. https://doi.org/https://doi.org/10.1016/j.foreco.2013.08.014

Raumonen, P., Åkerblom, M., Kaasalainen, M., & others. (2013). *TreeQSM: Quantitative structure models of trees from terrestrial laser scanning point clouds*. https://github.com/InverseTampere/TreeQSM.

Raumonen, Pasi, Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., & Lewis, P. (2013). Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sensing*, *5*(2), 491–520. https://doi.org/10.3390/rs5020491

Terryn, L., Calders, K., Disney, M., Origo, N., Malhi, Y., Newnham, G., Raumonen, P., Åkerblom, M., & Verbeeck, H. (2020). Tree species classification using structural features derived from terrestrial laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *168*, 170–181. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2020.08.009

Zhu, Z., Kleinn, C., & Nölke, N. (2020). Assessing tree crown volume—a review. *Forestry: An International Journal of Forest Research*, *94*(1), 18–35. https://doi.org/10.1093/forestry/cpaa037