treeX: Unsupervised Tree Instance Segmentation in Dense Forest Point Clouds

Josafat-Mattias Burmeister[a], Andreas Tockner[b], Stefan Reder Koithahn[c], Markus Engel[d], Rico Richter[a], Jan-Peter Mund[c], Jürgen Döllner[a,e]

[a] *University of Potsdam, Digital Engineering Faculty, Potsdam, Germany*
[b] *BOKU University, Department of Ecosystem Management, Climate and Biodiversity, Institute of Forest Growth, Wien, Austria*
[c] *Eberswalde University of Sustainable Development, Department of Forestry, Eberswalde, Germany*
[d] *State Forestry Research Centre, Eberswalde, Germany*
[e] *Hasso Plattner Institute, Potsdam, Germany*

1. Context and Objective

Close-range laser scanning enables the acquisition of detailed 3D point clouds of forest stands. Accurate tree instance segmentation is a prerequisite for deriving tree-level attributes such as stem diameters, tree heights, and crown architecture from these data. Although recent studies have proposed deep learning approaches for tree instance segmentation, such methods typically require large annotated datasets and substantial computational resources. As a resource-efficient alternative, we present an unsupervised method for tree instance segmentation in dense forest point clouds.

Our approach is a revised version of the treeX algorithm originally introduced by Tockner et al. (2022), which was designed primarily for personal laser scanning (PLS) data. We improve the algorithm's accuracy and extend its applicability by introducing two parameter presets: one for ground-based laser scanning (stationary terrestrial laser scanning - TLS and PLS) and one for UAV-borne laser scanning (ULS) data. To enable broader adoption, we provide an open-source Python implementation of the algorithm in the pointtree package.

2. TreeX Algorithm

The input of the treeX algorithm is a point cloud with xyz-coordinates and optional reflectance intensity values. It outputs a tree instance ID for each point, as well as estimates of tree position and stem diameter for each detected tree. The algorithm consists of three main steps: (1) First, a digital terrain model is constructed, and the height of each point above the terrain is computed. (2) Next, individual stems are detected by extracting a horizontal slice of the point cloud at a height range where stems are typically well separated. Points within this slice are clustered using DBSCAN, and false positive stem detections are removed through a series of filtering criteria. Stem positions and diameters are then estimated. (3) Finally, tree crowns are delineated using a region growing procedure. In this procedure, tree segments are initialized based on the detected stem positions and diameters and are iteratively expanded by incorporating neighboring points.

3. Evaluation

We evaluated our algorithm on six public datasets (FOR-instance V1, ForestSemantic, LAUTx, NIBIO MLS, TreeLearn, Wytham Woods), covering different sensor types and forest characteristics. As reference methods, we considered six open-source tree instance segmentation approaches, including algorithmic methods (original version of treeX, treeiso, and RayCloudTools) and deep learning methods (ForAINet, SegmentAnyTree, and TreeLearn).

4. Results and Discussion

Compared to the original treeX algorithm, our revision substantially improves accuracy for ground-based data while also reducing runtime in terms of wall-clock time (instance detection F_1-score gains of +0.11 (0.764 vs. 0.877) for LAUTx to +0.49 (0.052 vs. 0.547) for Wytham Woods; wall-clock time for Wytham Woods of 229 min vs. 140 min). For ULS data, our preset achieves an instance detection F_1-score of 0.58, whereas the original algorithm fails to segment any correct instances. Ablation studies indicate that these improvements are primarily due to refinements of the parameter settings and a more efficient circle fitting procedure in the stem detection step.

The accuracy of our algorithm is competitive with existing open-source tree instance segmentation methods (Fig. 1). Compared to other algorithmic approaches, our method demonstrates improved generalizability across datasets (mean instance detection F_1-score across all evaluation datasets: 0.68 for our algorithm, 0.5 for RayCloudTools, 0.47 for treeiso). Compared to deep learning approaches, the overall accuracy of our method is at a similar level (mean instance detection F_1-score across all evaluation datasets:

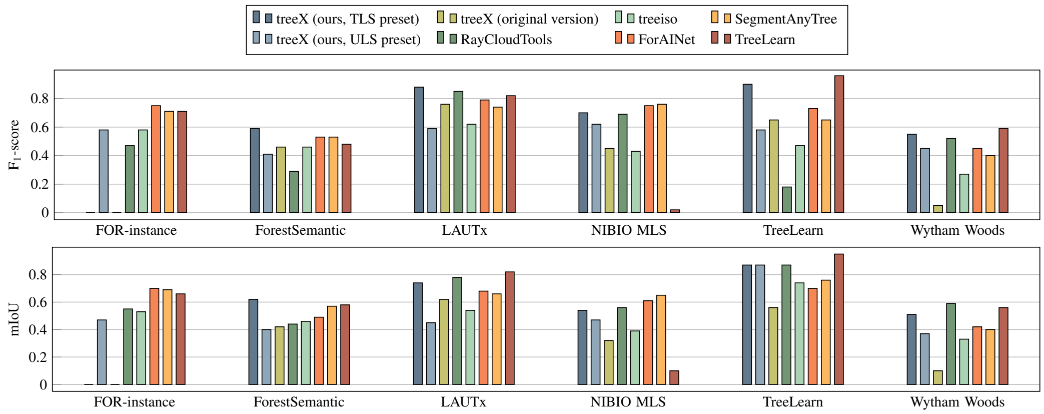

Figure 1: Accuracy comparison of our algorithm with existing open-source tree instance segmentation methods. The upper chart shows the instance detection F_1-score, and the lower chart the instance segmentation mean intersection over union (mIoU).

0.68 for our algorithm, 0.67 for ForAINet, 0.63 for SegmentAnyTree, 0.6 for TreeLearn). Notably, the performance of deep learning methods is strongly influenced by the composition of their training data, and these methods outperform our approach and the other tested algorithmic methods when applied to sparse point cloud data, such as the FOR-instance V1 dataset. In contrast, our method performs best on dense ground-based TLS and PLS scans, where stems are well represented. For ULS data, the sparser representation of below-canopy areas increases the likelihood of stem detection errors, thereby limiting the applicability of our approach to such data.

Consequently, our method is particularly valuable in the following scenarios: (1) as a resource-efficient alternative to deep learning approaches in scenarios where the data characteristics align with the method design (sufficient stem visibility and point density), and (2) for the semi-automatic generation of labeled training data for deep learning models, thereby supporting the development of more generalizable deep learning methods.

References

Tockner, A., Gollob, C., Kraßnitzer, R., Ritter, T., Nothdurft, A., 2022. Automatic tree crown segmentation using dense forest point clouds from personal laser scanning (PLS). International Journal of Applied Earth Observation and Geoinformation 114, 103025. doi:10.1016/j.jag.2022.103025.