BENCHMARKING INSTANCE SEGMENTATION IN TERRESTRIAL LASER SCANNING FOREST POINT CLOUDS

Wout Cherlet[1], Zane Cooper[1], Wouter A.J. Van den Broeck[1], Mathias Disney[2], Niall Origo[3], Kim Calders[1]

[1]CAVElab, Department of Environment, Ghent University, Ghent, Belgium;
[2]UCL Department of Geography, London, UK;
[3]Climate and Earth Observation group-National Physical Laboratory, Teddington, UK

ABSTRACT

Terrestrial laser scanning (TLS) has proven to be an invaluable tool in various forest ecology applications and forestry research. A crucial step in most TLS forest point cloud processing pipelines is instance segmentation; separating individual trees from the forest. However, automation in this area proves difficult, largely due to the heterogeneity of tree features and composition as well as overlapping, dense crown areas and understory. A lack of benchmarks and standard metrics complicates intercomparison of methods and hinders development in the field. This work proposes a set of metrics and methodology for benchmarking methods, and applies this to four open source TLS instance segmentation methods on a fully segmented 1.2 hectare benchmark dataset of a deciduous forest.

Index Terms— Forest point clouds, Instance segmentation, Terrestrial laser scanning

1. INTRODUCTION

Terrestrial laser scanning (TLS) is an active remote sensing technology used to create a highly detailed three-dimensional representation of the surrounding area. This technology was first introduced in the early 2000s for basic forest inventory tasks, like tree height and diameter at breast height (DBH). Advancements in fieldwork protocols, sensor equipment and processing have made data collection of hectare-sized forest plots possible [1].

Forest point clouds acquired by TLS are used for a number of applications. In forest inventory and ecology, TLS point clouds provide access to a number of metrics, such as diameter at breast height (DBH), height or vertical crown projected area. Additionally, point cloud data opens a realm of measurements not accessible using conventional tools, essential for fundamental ecological understanding of forest ecosystems. Aboveground biomass (AGB), an important indicator of forest productivity, is classically estimated by measuring tree DBH and applying allometric scaling models. However, AGB estimation from TLS data has not only shown better agreement with destructive samples independent of DBH, but enables new insights into the distribution of biomass in forests [2]. Recently, TLS point cloud data has been used as structural input to build a virtual forest for use in radiative transfer (RT) modelling [3]. Radiometric properties are linked to each tree to obtain a structurally and radiometrically accurate digital twin.

A crucial step in the processing pipeline of TLS forest point clouds for these applications is instance segmentation, i.e. separating individual trees from the forest. This segmentation enables both explicit 3D modelling of trees by way of Quantitative Structure Models (QSMs), and more detailed traits to be extracted from the point cloud directly. However, instance segmentation remains the most challenging in automated processing pipelines [4]. Due to overlapping crowns, dense vegetation and/or varying growing patterns, manual intervention is often needed to resegment parts of the crown material or separate conjoined instances. Furthermore, a lack of standard benchmarks and evaluation practices hinders the development of new methods, as most developers evaluate new methods on a specific dataset, making intercomparison based on these results difficult. Recently, such a standardized benchmark alongside proposed evaluation methods was released for UAV-LS data [5], but no similar dataset exists for TLS data.

This work proposes a set of evaluation metrics and methodology to compare the performance of instance segmentation methods on TLS data. This methodology is applied to a number of open-source algorithmic and deep learning methods, using

a fully segmented benchmark dataset of 135 by 88 meter containing 783 tree instances, divided into training, validation and testing areas.

2. METHODS AND MATERIALS

2.1. Instance segmentation methods

Two distinct approaches to the instance segmentation problem can be discerned: classical rule-based algorithms, and the more recent development of deep learning based approaches. Algorithmic methods utilize domain-specific knowledge condensed into an exact, deterministic set of rules. Most of these algorithms start by identifying tree trunks and iteratively build up trees from there. On the other hand, deep learning methods are purely data-driven and attempt to learn tree features directly from labeled point clouds in a training set. The state of the art on urban and indoor benchmarks has long been dominated by deep learning based methods. However, only recently have such methods been applied in a forest setting.
Below, all methods used in the benchmark are described. Only open-source methods have been considered. Preprocessing requirements as described in the documentation have been followed as closely as possible. Unless mentioned otherwise, default parameters were used where applicable.

2.1.1. Algorithmic methods

RayCloudTools [6] is a set of command line tools aimed at processing rayclouds, point clouds augmented with their scanner position. Their tree extraction algorithm does not explicitly use the raycloud information however. Trees are reconstructed by calculating shortest paths from each point to the ground, and then an agglomeration of paths from the ground up.
Treeiso [7] is a three-step graph-based method based on the cut-pursuit algorithm. In the first and second step, the cut-pursuit algorithm is used first in 3D to segment the point cloud into small, coherent clusters. Subsequently, it is used in the 2D plane, with cluster centroids as nodes and inverse 3D cluster edge-to-edge distances as weights. In a final third step, over-segmented clusters are detected and merged to the nearest tree. Treeiso is the only algorithm in the benchmark that requires ground points be removed first. For this, the Cloth Simulation Filter (CSF) plugin for CloudCompare was used. The parameters of CSF were tuned extensively to minimize its effects on the instance segmentation results. For the second step, various parameter combinations were tried and visually evaluated, a K value and lambda value of 40 were used for the final results.

2.1.2. Deep learning methods

TreeLearn [8] is a two-step pipeline, using first a sparse 3D convolutional neural network to predict learnable offset vectors pointing towards the two-dimensional tree centers, and a semantic score to determine if points are part of a tree or the understory/ground. The second step utilizes a clustering algorithm on the points that are transformed based on their predicted offset vectors to identify instances. As required, the test area was buffered with a 10 meter unlabeled edge on each side, as to provide context for the offset prediction. This edge is not used during evaluation. The provided pre-trained checkpoints were fine-tuned for 1200 epochs.
In [9], Xiang et al. investigate different bottom-up approaches to large scale panoptic segmentation. They design different segmentation pipeline architectures, and evaluate them on the urban NPM3D and the FORinstance benchmark. Similar to TreeLearn, a 3D neural network is used to obtain per-point tree features to serve as input to three segmentation branches. The first performs semantic segmentation using a small neural network with a single hidden layer. The centre offset branch employs the same strategy as TreeLearn, aiming to predict for each point the offset vector to its instance center. A final instance embedding branch embeds each point in a 5D feature space optimised to separate instances spatially. Clustering is then applied to both branches, and their results are combined using a final small scoring network. Their experiments showed that the combination of these clustering strategies increased instance segmentation performance compared to only using one of the two. This setup was used to train a model for 150 epochs with batch size 8.

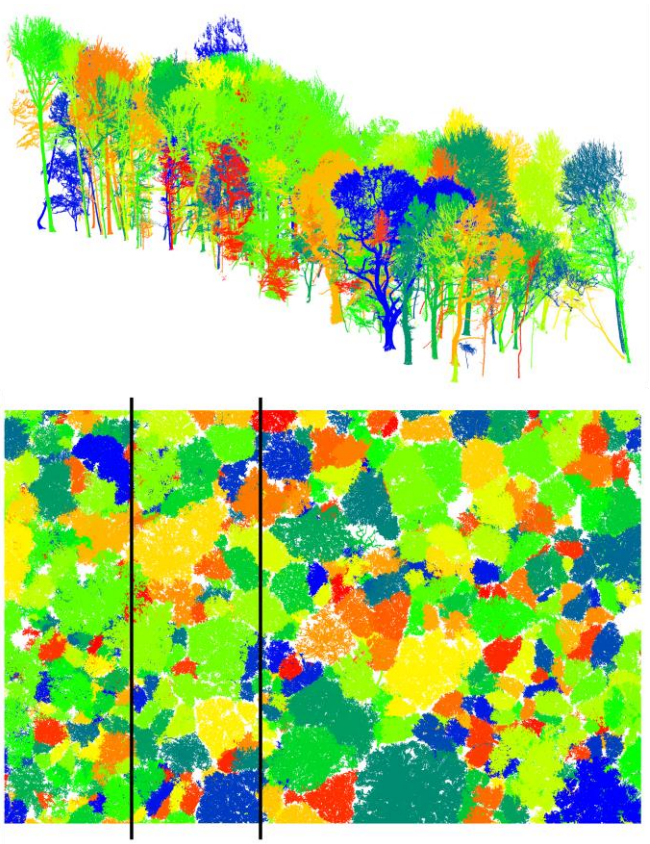

Fig. 1. Top: All tree instances in the testing area. Bottom: Top-down view of the testing, validation and training area respectively.

2.2. Data

Data was collected at the Wytham Woods study site using a RIEGL VZ-400 terrestrial laser scanning, during late November 2015 - January 2016 in leaf-off conditions [10]. Scans were done in a 6 ha area using an approximate 20 by 20 meter grid, and later downsampled to 0.026 cm resolution using voxel grid aggregation. Trees in a smaller 1.4 ha study area were initially segmented using the open-source software Treeseg [11], and further inspected and manually corrected by experts [3].

By subsetting a 135 by 88 meter rectangular point cloud out of the center of this study area, a fully segmented plot with all points either classified as a tree instance or understory/ground was created. Some very limited cleaning of misclassified points and branches in the canopy was performed. The rectangular study area was then divided into a test, validation and training area of 20 %, 20 % and 60 % of the study area each, by splitting the point cloud along the x-axis. The testing area and the test/validation/training split of the plot are shown on figure 1. The test area of around 27 by 88 meter, on which all evaluation metrics are calculated, contains a total of 245 trees. To limit border effects only trees that have at least 90 % of their points contained in this area are considered in these metrics, leaving 181 evaluation trees. This threshold was chosen to include some large trees that only have a small part of their canopy outside of the test areas, however 160 out of these 181 trees lie entirely in this test area. The training dataset used to train/fine-tune both deep learning models measures about 81 by 88 meters and contains 431 instances.

2.3. Evaluation metrics

The evaluation of the instance segmentation task can be performed on two major levels: the plot- and the instance level. At the plot level, we evaluate how well the plot is segmented overall. This is measured by precision, recall and F1 score:

$$Prec = \frac{TP}{TP + FP} \quad (1)$$

$$Rec = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 * \frac{Prec * Rec}{Prec + Rec} \quad (3)$$

In the definition of these metrics, True Positives (TP) describe a match between prediction and ground truth, and thus correctly predicted trees. False Positives (FP) are predictions that are not matched to ground truth trees and False Negatives (FN) are ground-truth trees that are not matched to any prediction. Precision thus tells us the fraction of predictions that are actual instances, while recall tells us the fraction of ground-truth instances that were detected. The F1-score is the harmonic mean of the 2, and is a metric of the overall performance of the method.

Tree predictions are matched to ground-truth instances by calculating the intersection over union (IoU); the number of points shared between both point clouds over the total number of shared and non-shared points. Then, Hungarian matching [12] is performed to generate the most effective prediction-ground truth pairs. A prediction is then considered a TP if its IoU with the matched ground-truth is over 0.5, otherwise the prediction will be deemed a FP and the ground-truth instance a FN. As the test area contains fractions of instances that can be hard to detect, trees that have less than 90 % of their points in the test area are not counted as FN. Similarly, predictions that match best with one of these instances are not taken into account as either TP or FP, as to not let border effects affect these metrics.

On the other hand, at the instance level, we evaluate how well individual trees are segmented. Again, we can use the precision, recall and F1 score for this, now defined at the point scale. A TP point is a point both present in ground truth tree and prediction, a FP point a point only present in the prediction and a FN point is a point only present in the ground truth. Here, precision thus tells us the fraction of predicted points that are also present in the ground truth instance. On the other hand, recall shows the fraction of ground truth points present in the prediction. The precision, recall and F1 score are then averaged over all successfully matched predictions to produce the mPrec, mRec and mF1 metrics. Additionally, we average the IoU of each matched tree with its prediction to produce the mIoU metric.

3. RESULTS AND DISCUSSION

The benchmarking results are summarized in table 1, showing both plot- and tree-level metrics. Additionally, figure 2 shows segmentation results on the tree with the largest number of points in the test dataset.

RayCloudTools shows the best performance for all but one, detecting about 70 % of the trees in the plot with 50 % precision. Visual inspection of the results show that in general methods tend to oversegment larger trees and fuse smaller trees together. Due to the IoU threshold of 0.5, which is commonly used as the lowest acceptable threshold for segmentation tasks, oversegmented trees and fused instances additionally lead to missed detections and thus lower recall.

The two deep learning methods included in the benchmark show promising results. While Xiang et al. [9] concluded that using both a centre offset and instance embedding branch before clustering, this conclusion is not reflected in the results, performing slightly worse then the TreeLearn model. This discrepancy is likely due to the pre-training of the TreeLearn model on beech-dominated TLS and mobile laser scanning (MLS) point clouds. Producing fully segmented datasets of the size and quality like the one used in this work requires a lot of precise manual work. However, despite the fairly large training area, the number of trees is still too few to effectively train the large feature backbones that perform the feature extraction in these methods. Therefore, pre-training this backbone using unsupervised learning tasks such as occlusion completion, where practically unlimited training data can be generated, or using synthetic training data is likely to significantly improve results.

One of the main roadblocks in standardizing benchmarks is the huge variety in forest and data types. This work focused on TLS data of deciduous forest, but generalization to MLS and UAV-LS data and other forest types is difficult. While efforts like FORinstance are a step in the right direction, the lack of fully segmented large-area plots in different settings hinders training and evaluation of methods on a broader scale.

Table 1. Benchmark results for all methods.

	Plot-level metrics			Tree-level metrics			
	Precision	Recall	F1	mPrec	mRec	mF1	mIoU
RayCloudTools	0.465	0.702	0.559	0.914	0.931	0.915	0.854
TreeIso	0.410	0.276	0.330	0.822	0.888	0.832	0.722
TreeLearn	0.720	0.425	0.535	0.854	0.924	0.878	0.794
Xiang et al. [9]	0.506	0.431	0.466	0.850	0.881	0.849	0.751

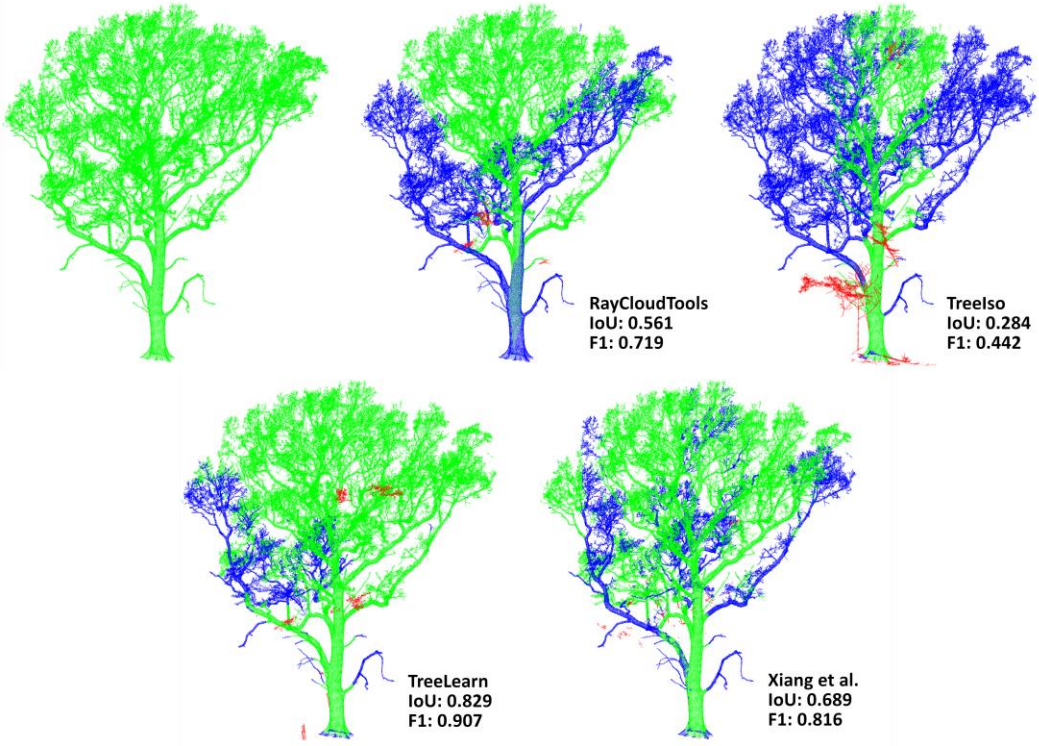

Fig. 2. Segmentation results on the largest tree in the dataset. TP points are shown in green, FP points in red and FN points in blue. Corresponding IoU and F1 score for each prediction is shown.

Finally, algorithms for post-processing are not investigated here but play a major role in obtaining accurate tree models. Such tasks include eliminating the effects of wind and noise, segmenting epiphytes and other plants from tree instances, and leaf-wood segmentation.

4. CONCLUSION

In this work, a number of tree instance segmentation approaches were compared using a proposed set of evaluation metrics on a fully segmented TLS forest point cloud dataset. Results show that instance segmentation remains a difficult task, with the best method detecting only 70 percent of trees with under 50 percent precision. Recent methods employing deep learning models for automatic tree feature extraction show promising results, even with a relatively small training dataset.

5. REFERENCES

[1] Kim Calders, Jennifer Adams, John Armston, Harm Bartholomeus, Sebastien Bauwens, Lisa Patrick Bentley, Jerome Chave, F Mark Danson, Miro Demol, Mathias Disney, Rachel Gaulton, Sruthi M Krishna Moorthy, Shaun R Levick, Ninni Saarinen, Crystal Schaaf, Atticus Stovall, Louise Terryn, Phil Wilkes, and Hans Verbeeck, "Terrestrial laser scanning in forest ecology: Expanding the horizon," *Remote Sens. Environ.*, vol. 251, pp. 112102, Dec. 2020.

[2] Kim Calders, Glenn Newnham, Andrew Burt, Simon Murphy, Pasi Raumonen, Martin Herold, Darius Culvenor, Valerio Avitabile, Mathias Disney, John Armston, and Mikko Kaasalainen, "Nondestructive estimates of above-ground biomass using terrestrial laser scanning," *Methods in Ecology and Evolution*, vol. 6, no. 2, pp. 198–208, 2015.

[3] Kim Calders, Niall Origo, Andrew Burt, Mathias Disney, Joanne Nightingale, Pasi Raumonen, Markku Akerblom, Yadvinder Malhi, and Philip Lewis, "Real-˚ istic forest stand reconstruction from terrestrial lidar for radiative transfer modelling," *Remote Sensing*, vol. 10, no. 6, 2018.

[4] Olivier Martin-Ducup, II Mofack, Gislain, Di Wang, Pasi Raumonen, Pierre Ploton, Bonaventure Sonke,´ Nicolas Barbier, Pierre Couteron, and Raphael P¨ elissier,´ "Evaluation of automated pipelines for tree and plot metric estimation from TLS data in tropical forest areas," *Annals of Botany*, vol. 128, no. 6, pp. 753–766, 04 2021.

[5] Stefano Puliti, Grant Pearse, Peter Surovy, Luke Wal-´ lace, Markus Hollaus, Maciej Wielgosz, and Rasmus Astrup, "FOR-instance: a UAV laser scanning benchmark dataset for semantic and instance segmentation of individual trees," Sept. 2023.

[6] Thomas D. Lowe and Kazys Stepanas, "Raycloudtools: A concise interface for analysis and manipulation of ray clouds," *IEEE Access*, vol. 9, pp. 79712–79724, 2021.

[7] Zhouxin Xi and Chris Hopkinson, "3d graph-based individual-tree isolation (treeiso) from terrestrial laser scanning point clouds," *Remote Sensing*, vol. 14, no. 23, 2022.

[8] Jonathan Henrich, Jan van Delden, Dominik Seidel, Thomas Kneib, and Alexander Ecker, "Treelearn: A comprehensive deep learning method for segmenting individual trees from forest point clouds," 2023.

[9] Binbin Xiang, Torben Peters, Theodora Kontogianni, Frawa Vetterli, Stefano Puliti, Rasmus Astrup, and Konrad Schindler, "Towards accurate instance segmentation in large-scale lidar point clouds," 2023.

[10] Kim Calders, Hans Verbeeck, Andrew Burt, Niall Origo, Joanne Nightingale, Yadvinder Malhi, Phil Wilkes, Pasi Raumonen, Robert Bunce, and Mathias Disney, "Terrestrial laser scanning data Wytham Woods: individual trees and quantitative structure models (QSMs)," Nov. 2022.

[11] Andrew Burt, Mathias Disney, and Kim Calders, "Extracting individual trees from lidar point clouds using treeseg," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 438–445, 2019.

[12] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.