A Persistence-Based Approach for Individual Tree Mapping

Xin Xu
The University of Maryland at College Park
College Park, Maryland, USA
xinxu629@umd.edu

Federico Iuricich
Clemson University
Clemson, South Carolina, USA
fiurici@clemson.edu

Leila De Floriani
The University of Maryland at College Park
College Park, Maryland, USA
deflo@umiacs.umd.edu

ABSTRACT

Light Detection and Ranging (LiDAR) sensors generate dense point clouds that can be used to map forest structures at a high spatial resolution level. In this work, we consider the problem of identifying individual trees in a LiDAR point cloud. Existing techniques generally require intense parameter tuning and user interactions. Our goal is defining an automatic approach capable of providing robust results with minimal user interactions.

To this end, we define a segmentation algorithm based on the watershed transform and persistence-based simplification. The proposed algorithm uses a divide-and-conquer technique to split a LiDAR point cloud into regions with uniform density. Within each region, single trees are identified by applying a segmentation approach based on watershed by simulated immersion. Experiments show that our approach performs better than state-of-the-art algorithms on most of the study areas in the benchmark provided by the *NEW technologies for a better mountain FORest timber mobilization* (NEWFOR) project. Moreover, our approach requires a single (Boolean) parameter. This makes our approach well suited for a wide range of forest analysis applications, including biomass estimation, or field inventory surveys.

CCS CONCEPTS

• **Computing methodologies** → **Shape analysis**; • **Information systems** → **Geographic information systems**.

KEYWORDS

LiDAR, Tree segmentation, Watershed transform, Topological persistence

ACM Reference Format:
Xin Xu, Federico Iuricich, and Leila De Floriani. 2020. A Persistence-Based Approach for Individual Tree Mapping. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397536.3422231

1 INTRODUCTION

Identifying individual trees composing a forest is a crucial step for characterizing forest structures and to forecast their changes [16]. Historically, forest inventories were completed manually by expensive and time-consuming field surveys [21]. Light Detection and Ranging (LiDAR) technologies, such as Airborne Laser Scanning (ALS), provided new efficient ways for such inventories. Still, identifying single tree structures presents multiple challenges due to noise, occlusions, and the diverse structure characterizing different types of forest.

In the last decade, different methods have been developed for segmenting ALS point clouds. According to Koch et al. [12], these can be classified as: *raster-based*, *point-based*, *or combined strategies*.

Raster-based methods are by far the most common type of approach. The idea at the base of these methods is that of computing and segmenting a raster product generated from points such as the Canopy Height Model (CHM) [3, 10, 18]. Raster-based methods force the discretization of data points on a grid which can cause a loss of information especially in those area where point density is low. *Point-based methods* have been introduced to overcome this limitation by working on the input point cloud directly. Point-based methods identify single trees by analyzing the geometric structure of the point cloud by means of clustering techniques such as k-means [9], or mean-shift [7, 8]. Both raster-based and point-based methods often fail in identifying smaller trees covered by the crown of higher trees. To this end, *combined methods* have been developed by combining the strengths of raster and point-based approaches [2, 5, 17].

Despite the effort, a general approach capable of producing satisfactory results on a large variety of forests is still lacking. Moreover, state-of-the-art approaches make use of many parameters that a user needs to fine-tune in time-consuming trial and error phases.

We propose a new approach for individual tree segmentation requiring minimal user interactions. The key idea is to create two distinct segmentations of the forest point cloud. One segmentation aims at identifying single trees focusing on treetops. The second segmentation identifies single trees based on their bottoms (i.e., tree trunk). All parameters used by our approach are automatically tuned by the algorithm and only one Boolean input is needed from the user to produce the final segmentation.

We evaluate our approach on 14 study areas released in the *NEW technologies for a better mountain FORest timber mobilization* (NEWFOR) benchmark [6]. Our approach outperform state-of-the-art algorithms on 10 over 14 study areas while achieving comparable performances on the remaining 4.

2 BACKGROUND

In this work we consider a point cloud P as a collection of points in the Euclidean space $\mathbb{R}^3$. Points in P are connected by means of a graph $G = (N, A, f)$. Each node in N corresponds to a point in P. With abuse of notation we will indicate with p both a point and its

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3422231

corresponding node. A is the set of arcs connecting pairs of nodes in N. f is a scalar function defined on the nodes in N. We say that two nodes are *adjacent* if they share an arc. We call *neighbors* of a node p the nodes adjacent to p.

The watershed algorithm used in this work is based on the *discrete topographic distance* [14]. Discrete topographic distance is defined in terms of a minimum-cost path in a graph G. Given a node p in N, the *lower slope* $LS(p)$ is the maximal slope among the arcs incident in p: $LS(p) = \max\{\frac{f(p)-f(q)}{dist(p,q)} | (p,q) \in A, f(q) < f(p)\}$, where $dist(p,q)$ denotes the Euclidean distance between points p and q. The π-topographic distance between two nodes p and q is the sum of the costs of all directed arcs in the path π connecting p to q. The discrete topographic distance $T(p,q)$ between p and q is the minimum of the π-topographic distances along all such paths. The *catchment basin* of a minimum m of f is the set of nodes closer to m (in terms of topographic distance) than any other minimum of f. The *watershed nodes* are those nodes equally distant to more than one catchment basin.

3 INDIVIDUAL TREE SEGMENTATION

The proposed approach is organized in two phases. First, the input *forest point cloud* P is divided into clusters, that we call *tree clusters*. Second, each cluster is processed independently to identify all *tree point cloud* (i.e., group of points identifying each tree).

Phase 1. A preprocessing step is used to clean the input forest point cloud P from ground points and noise [20].

After the preprocessing step, the objective is to subdivide a group of trees that are easily separable, i.e., well-distanced trees should end up in different clusters while trees with intersecting canopy should end up in the same tree cluster.

Then, a graph $G_\epsilon = (N, A, f_d)$ is computed with the nodes N being the points in the point cloud. To compute the arcs in A we first project the point cloud on the x-y plane and we connect all points closer than a specified threshold ϵ. We describe in Section 4 how ϵ is automatically set by the algorithm. f_d is a scalar function, defined for each node in N, approximating the point density. Given a node $p \in N$, we define function

$$f_d(p) = \frac{\sum_{i=0}^{n} dist(p, p_i)}{n}$$

where p_i, with $0 \leq i \leq n$ are the points adjacent to p in N and $dist(p, p_i)$ is the Euclidean distance between two points. Intuitively, this associates low values to points in regions of high density and high values to points in regions of low density.

Performing watershed segmentation on graph G_ϵ will provide a subdivision of G_ϵ in regions of influence of the minima of f_d. Notice that function f_d is noisy which results in a high number of minima and, consequently, a high number of clusters. However, the boundaries between many trees are already well separated by regions of high values (i.e., low density).

Then, a persistence-based simplification [4] is used to reduce over segmentation. We assign a persistence value to each edge e as:

$$f_p(e) = \min(f_d(e) - f_d(p_1), f_d(e) - f_d(p_2)),$$

where p_1 and p_2 are the minima originating the two basins connected by e. Intuitively, each persistence value represents the inverse likelihood of two regions to be generated by noise. If $f_p(e)$ is low, the two regions are likely generated by noise and should be merged together.

Our algorithm loops over the edges in increasing order of persistence. Given edge e and p_1 and p_2 the two basins connected by e, we merge p_1 and p_2 if $f_p(e) \leq \min_p + (\max_p - \min_p) * \theta$, where $\min_p$ and $\max_p$ are the minimum and maximum persistence values found, and θ is automatically set by our algorithm (see Section 4).

Once the tree clusters have been computed, points are re-projected in $\mathbb{R}^3$ (Figure 1), where each cluster will be processed independently.

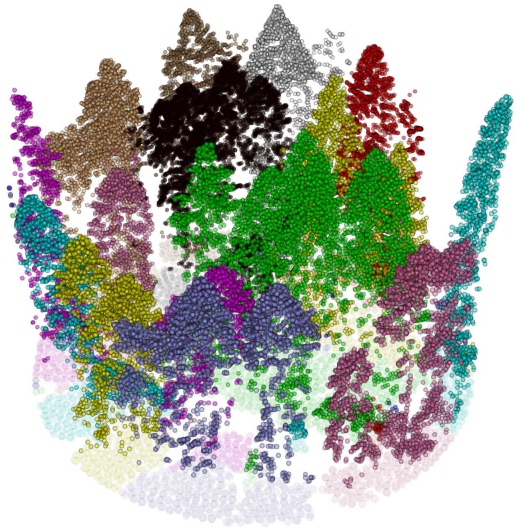

Figure 1: Tree clusters computed by the proposed approach.

Phase 2. The second phase consists of identifying single trees. This time, the point distribution is analyzed in full by considering positions of points in the three-dimensional space.

Our algorithm uses the height function and its inverse to create two distinct segmentations of a tree cluster. Single trees will be recognized by intersecting the two segmentations. The following steps are performed for each tree cluster.

We define a graph $G_c = (N, A, f)$. The nodes N are the points in the tree cluster. To compute the arcs in A we perform a neighbor search, for each point p, by using a cylinder positioned on the top of p. We fit a circle with center in p and radius r, and we extrude a cylinder from the circle with a height h. All points falling into the cylinder get connected to p. Section 4 describes how r and h are automatically selected.

After defining a topology for a tree cluster, we use watershed segmentation on G_c assuming f to be the height function. We can expect noise and small perturbations to create spurious regions that are removed by performing a merging operation based on persistence.

Each region has a minimum p associated with it. For each pair of adjacent regions m_1, m_2, we retrieve the points p_1 and p_2 originating m_1 and m_2, respectively. Moreover we retrieve the edge e, on the boundary of both m_1 and m_2, having minimum function

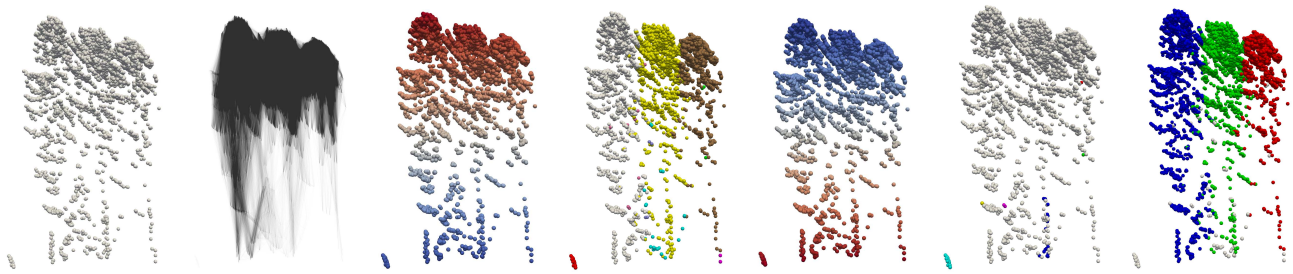

Figure 2: Retrieving individual trees from tree cluster points performed by the proposed approach. (a) Tree cluster points, (b) network constructed from tree cluster points, (c) points colored based on height according to a blue-red diverging color map, and segmentation $S_{\uparrow}$, (d) points colored as the inverse height function, and segmentation $S_{\downarrow}$, (e) final segmentation obtained by intersecting segmentations in (c) and (d).

value. Then, we associate a persistence value to the pair m_1, m_2, as $\min_{i=\{1,2\}} |(f(p_i) - f(e))|$.

A persistence-based simplification is performed by merging regions having normalized persistence lower than a preset threshold of θ. After that, we obtain a segmentation $S_{\uparrow}$ based on the height function. The same process is repeated by using the inverse of the height function. Specifically, we run the same process on G_c by using $\frac{1}{f}$ instead of f, which produces a second segmentation $S_{\downarrow}$.

An example of the results obtained is shown in Figure 2. Figure 2(c) shows the height function, color-coded on the tree cluster according to a blue-red diverging color map, and the segmentation $S_{\uparrow}$ obtained with our approach. As we notice, the three tree bottoms are well defined in the original point cloud which results in the correct segmentation of the three trees. Figure 2(d) shows the inverse of the height function and the segmentation $S_{\downarrow}$. In this case, the treetops are not distinct which results in the creation of a single big segment including all tress.

The last step consists of creating the final segmentation S_F by performing an intersection of the two segmentations $S_{\uparrow}$ and $S_{\downarrow}$.

At this stage, each point in the point cloud has a pair of labels associated with it, one defined by $S_{\uparrow}$ and the other defined by $S_{\downarrow}$. Then, a region in S_F is defined by the set of points having the same pair of labels $S_{\uparrow}$ and $S_{\downarrow}$. S_F is shown in Figure 2(e) which includes three trees.

4 PARAMETER SETTING

The main objective of our proposed method is that of requiring minimal user interactions. The user is asked for a single input which is the definition of the *dominant* segmentation. The user has to select which segmentation, between $S_{\downarrow}$ and $S_{\uparrow}$, is more likely to identify single trees. The choice should be done based on the type of trees in the forest. In the case of dominant conifer-like trees, $S_{\downarrow}$ should be the dominant segmentation since the treetops are easier to identify. Otherwise, if tree trunks are easier to spot, $S_{\uparrow}$ should be selected so that tree bottoms will drive the overall result.

Once the dominant segmentation is selected, all remaining parameters are automatically set as follows.

Distance ϵ: it is used to construct graph G_ϵ. We start by computing the graph G_ϵ for $\epsilon = 0.5$ and we use an iterative approach for tuning ϵ based on the average valence of the nodes of G_ϵ. If the average valence is between 150 and 50 we select current ϵ. If the valence is greater than 150, ϵ is decreased by 0.25. Otherwise, it is increased by 0.25.

Cylinder radius r and height h: they define the shape of the cylinder used to connect points in phase 2. Radius r is set equal to ϵ. Height h is computed as half the elevation of the highest point in the tree cluster.

Persistence-based simplification θ: it defines the amount of noise-removal operations performed while retrieving individual trees. Notice that we use two independent values for θ. One is used when working on segmentation $S_{\uparrow}$, and a different value is used when working on $S_{\downarrow}$. θ can assume any value between 0 and 1, where 0 indicates no noise-removal operation while 1 indicates that all regions are merged. The rationale is that the dominant segmentation, chosen by the user, should contain a smaller amount of noise. Then, for this segmentation, we set $\theta = 0.1$, which corresponds to a light noise-removal. The other segmentation instead is assumed to be noisier. We set $\theta = 0.7$ for a strong noise-removal.

5 EXPERIMENTAL RESULTS

In this section we discuss the results obtained with the proposed approach. We have implemented our algorithm as a plugin for the Topological Toolkit [19] which is an open-source library for topological data analysis and visualization. The output segmentations produced by our plugin are rendered in Paraview [1]. Experiments are run on a desktop computer equipped with a Intel ® Core™ CPU i7-8700 @ 3.20GHz and 32GB memory of RAM.

Experiments are performed on the *NEW technologies for a better mountain FORest timber mobilization* (NEWFOR) dataset [6]. The dataset consists of 14 publicly available study areas (i.e., ALS point cloud data) with associated field survey data. Field data include tree height, location, and Diameter at Breast Height (DBH) for each tree. Study areas are predominantly populated with fir, spruce, and pines.

We compare our approach with four methods provided by the publicly available lidR package [11]. Dalponte et al. [3], Silva et al. [18] and Watershed [15] are raster-based approaches. The fourth method, Li et al. [13] is a pure point-based approach. Parameters

for Li et al. [13] are set based on the suggested values in the original paper. Parameters for the remaining algorithms are tuned with a trial-and-error approach.

Comparisons. Considering timings, raster-based approaches take 5.88 seconds to run on average, except for one study area where they need around 20 seconds. Our method is faster on 8 over 14 study areas. The point-based approach by Li et al. [13] is the fastest overall, and this is four times faster, on average, compared to ours. It is important to remark that all methods, except ours, required additional time for tuning their parameters. This time is not counted in the comparison.

The accuracy of each method has been validated at the individual tree level. We replicate the same matching approach used by the NEWFOR benchmark [6], where segmented trees and trees in the field data are matched based on their location and their height. We report the *matching rate* calculated as the number of matched trees divided by the total number of trees in the field data. Overall, our method performs best on 10 over 14 study areas with an average matching rate of (38.39%). Our method outperforms other approaches in both high-density and low-density areas. For example, in a high-density area, our approach performs best with 48.98% matching rate. In low-density areas, the approach has 38.18% matching rate, which is 1.4 times the matching rate of any other method. In the 4 study areas where our method does not score best, the matching rate is comparable to those of other approaches, and segmentations look convincing under visual inspection. Notice that a direct comparison with the results of the NEWFOR benchmark is not feasible [6]. NEWFOR results have been evaluated on a cleaned-up version of the publicly available dataset where outliers and wrong data have been manually removed. Still, the average matching rate reported by the NEWFOR benchmark is 47%, which is close to the average matching rate of 38.39% we achieve on the "noisy" dataset.

6 CONCLUDING REMARKS

We have proposed a new robust approach for individual tree segmentation. We have compared our approach with state-of-the-art methods, and we proved that it provides improved accuracy in single tree identification. Additionally, our approach requires no parameter tuning, which makes it better suited for a large scale forest analysis.

A parallel version of our algorithm is under development with the scope of improving performances. We are currently testing our approach using different types of forest. Aside from ALS data, we are interested in applying our approach to other types of LiDAR data, such as Terrestrial LiDAR.

ACKNOWLEDGMENTS

We thank Ralph Dubayah, Hao Tang, Laura Duncanson, John Armston and Steven Hancock for their help and valuable comments. This work has been partially supported by the US National Science Foundation under grant number IIS-1910766.

REFERENCES

[1] Utkarsh Ayachit. 2015. *The ParaView Guide: Updated for ParaView Version 4.3* (full color version ed.). Kitware, Los Alamos.

[2] Elias Ayrey, Shawn Fraver, John A Kershaw Jr, Laura S Kenefic, Daniel Hayes, Aaron R Weiskittel, and Brian E Roth. 2017. Layer stacking: a novel algorithm for individual forest tree segmentation from LiDAR point clouds. *Canadian Journal of Remote Sensing* 43, 1 (2017), 16–27.

[3] Michele Dalponte and David A Coomes. 2016. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods in ecology and evolution* 7, 10 (2016), 1236–1245.

[4] Leila De Floriani, Ulderico Fugacci, Federico Iuricich, and Paola Magillo. 2015. Morse complexes for shape segmentation and homological analysis: discrete models and algorithms. In *Computer Graphics Forum*, Vol. 34. Blackwell Publishing Ltd, 761–785. https://doi.org/10.1111/cgf.12596

[5] Laura Duncanson, Bruce Cook, George Hurtt, and Ralph Dubayah. 2014. An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems. *Remote Sensing of Environment* 154 (2014), 378–386.

[6] Lothar Eysn, Markus Hollaus, Eva Lindberg, Frédéric Berger, Jean-Matthieu Monnet, Michele Dalponte, Milan Kobal, Marco Pellegrini, Emanuele Lingua, Domen Mongus, et al. 2015. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space. *Forests* 6, 5 (2015), 1721–1747.

[7] António Ferraz, Frédéric Bretar, Stéphane Jacquemoud, Gil Gonçalves, Luisa Pereira, Margarida Tomé, and Paula Soares. 2012. 3-D mapping of a multi-layered Mediterranean forest using ALS data. *Remote Sensing of Environment* 121 (2012), 210–223.

[8] António Ferraz, Sassan Saatchi, Clément Mallet, and Victoria Meyer. 2016. Lidar detection of individual tree size in tropical forests. *Remote Sensing of Environment* 183 (2016), 318–333.

[9] Sandeep Gupta, Holger Weinacker, and Barbara Koch. 2010. Comparative analysis of clustering-based approaches for 3-D single tree detection using airborne fullwave lidar data. *Remote Sensing* 2, 4 (2010), 968–989.

[10] Juha Hyyppa, Olavi Kelle, Mikko Lehikoinen, and Mikko Inkinen. 2001. A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. *IEEE Transactions on Geoscience and remote Sensing* 39, 5 (2001), 969–975.

[11] Roussel Jean-Romain, Auty David, De Boissieu Florian, and Sánchez Meador Andrew. 2020. *lidR: Airborne LiDAR Data Manipulation and Visualization for Forestry Applications.* https://cran.r-project.org/package=lidR R package version 2.2.4.

[12] Barbara Koch, Teja Kattenborn, Christoph Straub, and Jari Vauhkonen. 2014. Segmentation of forest to tree objects. In *Forestry Applications of Airborne Laser Scanning*. Springer, 89–112.

[13] Wenkai Li, Qinghua Guo, Marek K Jakubowski, and Maggi Kelly. 2012. A new method for segmenting individual trees from the lidar point cloud. *Photogrammetric Engineering & Remote Sensing* 78, 1 (2012), 75–84.

[14] Alan Mangan and Ross Whitaker. Oct.-Dec./1999. Partitioning 3D Surface Meshes Using Watershed Segmentation. *IEEE Transactions on Visualization and Computer Graphics* 5, 4 (Oct.-Dec./1999), 308–321. https://doi.org/10.1109/2945.817348

[15] Gregoire Pau, Florian Fuchs, Oleg Sklyar, Michael Boutros, and Wolfgang Huber. 2010. EBImage—an R package for image processing with applications to cellular phenotypes. *Bioinformatics* 26, 7 (2010), 979–981. https://doi.org/10.1093/bioinformatics/btq046

[16] Sorin C Popescu. 2007. Estimating biomass of individual pine trees using airborne lidar. *Biomass and Bioenergy* 31, 9 (2007), 646–655.

[17] Josef Reitberger, Cl Schnörr, Peter Krzystek, and Uwe Stilla. 2009. 3D segmentation of single trees exploiting full waveform LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64, 6 (2009), 561–574.

[18] Carlos A Silva, Andrew T Hudak, Lee A Vierling, E Louise Loudermilk, Joseph J O'Brien, J Kevin Hiers, Steve B Jack, Carlos Gonzalez-Benecke, Heezin Lee, Michael J Falkowski, et al. 2016. Imputation of individual Longleaf Pine (Pinus palustris Mill.) Tree attributes from field and LiDAR data. *Canadian journal of remote sensing* 42, 5 (2016), 554–573.

[19] Julien Tierny, Guillaume Favelier, Joshua A. Levine, Charles Gueunet, and Michael Michaux. 2018. The Topology ToolKit. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 832–842. https://doi.org/10.1109/TVCG.2017.2743938

[20] Wuming Zhang, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xie, Xiaoyan Wang, and Guangjian Yan. 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing* 8, 6 (2016), 501.

[21] Zhen Zhen, Lindi J Quackenbush, and Lianjun Zhang. 2016. Trends in automatic individual tree crown detection and delineation—evolution of lidar data. *Remote Sensing* 8, 4 (2016), 333.