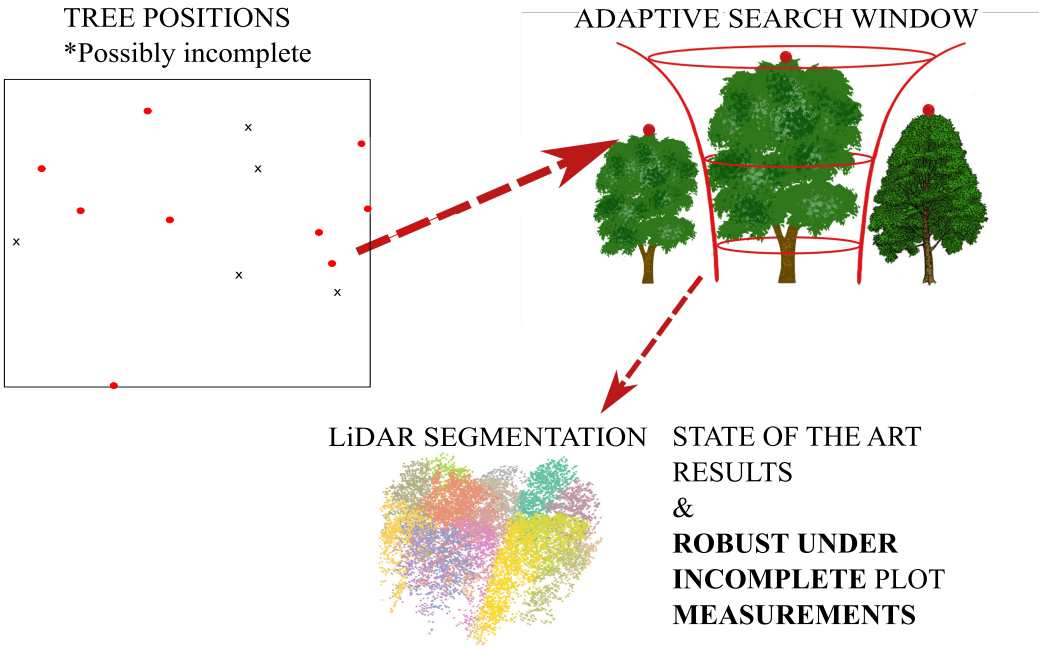
Adaptive Local Maxima Windows for Tree Segmentation: A Point Process Perspective

Konstantinos Florakis, Véronique Letort, Raphaël Canals, Gilles Faÿ, Samis Trevezas

Graphical Abstract

Adaptive Local Maxima Windows for Tree Segmentation: A Point Process Perspective

Konstantinos Florakis, Véronique Letort, Raphaël Canals, Gilles Faÿ, Samis Trevezas



Highlights

Adaptive Local Maxima Windows for Tree Segmentation: A Point Process Perspective

Konstantinos Florakis, Véronique Letort, Raphaël Canals, Gilles Faÿ, Samis Trevezas

- The proposed methodology achieves state-of-the-art results in LiDAR-based Individual Tree Segmentation
- Spatial point process theory guides robust Local Maxima window size selection
- Adaptive window approach preserves segmentation reliability despite incomplete plot inventories

Adaptive Local Maxima Windows for Tree Segmentation: A Point Process Perspective

Konstantinos Florakis[a,c,d,*], Véronique Letort[c], Raphaël Canals[a], Gilles Faÿ[c], Samis Trevezas[b,c]

[a] *University of Orléans, Château de la Source, Orléans, 45100, France*
[b] *Department of Mathematics, National and Kapodistrian University of Athens, Panepistimiopolis, Athens, 15784, Greece*
[c] *CentraleSupélec - Université Paris-Saclay, 3 Rue Joliot Curie, Gif-sur-Yvette, 91190, France*
[d] *Cybeletech, 2 Rue Chaintron, Montrouge, 92120, France*

Abstract

The growing accessibility of Light Detection And Ranging (LiDAR) data brings out novel perspectives that are crucial for tracking forest growth and enhancing resource management amid climate change. Utilizing these data to propose decision-support tools involves a vital step of segmenting individual trees. A widely adopted class of methods for this step is known as Local Maxima algorithms, which, although unsupervised, rely on per-site and/or per-species hyperparameter tuning for optimal performance. In this work, we introduce a novel theoretical framework grounded in point process theory to jointly model the data generation process and provide formal implementation guidelines for refining window size selection within the class of Local Maxima algorithms. This methodology can also be applied to incomplete plot measurements, a constraint noted in most data acquisition procedures. To ensure the reproducibility of the results and validate the practical application, we apply the proposed methodology in two cases: (i) a simulated dataset (made publicly available) and (ii) an open dataset. The method outperforms the alternative methodologies used as baselines in the simulation study, and achieves F_1-scores in the $55-90\%$ range on real data, depending on the positioning of the tree within the canopy relative to its neighboring trees.

Keywords: Spatial point processes, LiDAR, Segmentation, ITD, Hard-core point process

1. Introduction

In the rapidly evolving domain of forest monitoring and management, the advent of Light Detection and Ranging (LiDAR) systems has marked a transformative leap Tonolli et al. (2011); White et al. (2013). These sophisticated technologies facilitate high-resolution, three-dimensional representations of forested landscapes, offering unprecedented detail and accuracy (White et al., 2017). The ability to accurately seg-

*Corresponding author: konstantinos.florakis@centralesupelec.fr

ment individual trees from LiDAR data is pivotal for enhancing our understanding of forest structure and dynamics and for implementing effective conservation and management practices (Wang et al., 2016). However, this task presents unique challenges due to the intricate nature of forest canopies and the voluminous data generated by LiDAR systems (Tonolli et al., 2011; Wang et al., 2016, 2019). Furthermore, while current methodologies produce globally satisfactory results for dominant trees, with precision greater than 85% in Wang et al. (2016), for most methodologies and plot types, studies such as Kaartinen et al. (2012); Wang et al. (2016, 2019) report the need for algorithms capable of also satisfactorily segmenting trees of the lower canopy layers. These include the canopy classes of intermediate and suppressed trees, partially and completely overshadowed, respectively. For instance, Wang et al. (2016) tested segmentation algorithms under mixed forest conditions and different stages of development. Their study reported precision of $80-95\%$ for dominant trees and $70-90\%$ for co-dominant trees, while the corresponding values for intermediate and suppressed trees ranged between $20-70\%$ and $5-30\%$, respectively. Yancho et al. (2019) tested algorithms developed in recent years by Li et al. (2012), Dalponte and Coomes (2016), and Silva et al. (2016), while further introduced segmentation methods based on the color values of the points represented by their RGB components. They also stratified their results per canopy class, reporting precision of 80% for dominant trees, $55-60\%$ for co-dominant trees, and values smaller than 20% for the other classes.

Many segmentation algorithms incorporate an initial identification step of tree tops. One widely adopted method for this purpose is the Local Maxima (LM) method (see Chehreh et al. (2023) for a recent review). For clarity, the term *LM method* refers to the core algorithmic principle (subroutine) and not to the broader class of methods derived from or extending this approach. The LM method relies on a peak detection algorithm, selecting data points higher than their neighbors and meeting additional specific criteria (e.g. exceeding a certain threshold value). Methods based on LM appear in international benchmark studies such as Vauhkonen et al. (2012); Kaartinen et al. (2012); Eysn et al. (2015); Wang et al. (2016); Yancho et al. (2019) that have played a crucial role in evaluating and comparing the effectiveness of segmentation algorithms across different conditions. The performance of LM-based methods consistently approached or matched the highest achieved under the selected comparison criteria. In this context, improving the LM method represents a promising avenue of research for better segmentation performance in many application settings.

More specifically, applying the LM method requires specifying a search window and defining the neighboring envelope as a search space for the local maxima. This factor emerges as an important determinant of the method's performance, although there is no consensus on how to determine it. The associated window size can be either fixed (stable) or variable, i.e., *adjusted* with respect to (w.r.t.) tree/forest attributes, such as height, species, or crown diameter (Monnet et al., 2010).

- **Stable Window**: Stable window sizes remain widely used in LiDAR data

analysis for tree segmentation, and finding criteria for their optimization is an active area of research. Hyyppa et al. (2001) utilized a fixed $3\,\mathrm{m} \times 3\,\mathrm{m}$ square window, demonstrating the effectiveness of a non-adjustable approach in specific contexts. In 2006, Solberg et al. (2006) investigated the performance of small square windows, with sides ranging from 30 to 90 cm, reporting more accurate results for the smallest window size. Their approach included an adjustment of the Canopy Surface Model (CSM) using height residuals. Wallace et al. (2014b) have adapted the method from Reitberger et al. (2009) for tree segmentation using unmanned aerial vehicle (UAV) laser scanning data. They employ LM search over a Gaussian smoothed Canopy Height Model (CHM), detecting initial stem locations from local maxima within a $3\,\mathrm{m} \times 3\,\mathrm{m}$ window, a size optimized to have on average 2 points, focusing on points above the 65th percentile in above-ground height. In a related work, Wallace et al. (2014a) apply Point Cloud Detection and Delineation with a 2 m radius over a CHM smoothed again with a Gaussian filter. In more recent years, Gao et al. (2022) compared different window sizes of the form $i\,\mathrm{m} \times i\,\mathrm{m}$ for $i = 3, \ldots, 15$, finding smaller windows more effective for certain species, while Ma et al. (2022) assessed window sizes of $3\,\mathrm{m} \times 3\,\mathrm{m}$, $6\,\mathrm{m} \times 6\,\mathrm{m}$ and $9\,\mathrm{m} \times 9\,\mathrm{m}$ for segmenting Eucalyptus trees, concluding that the window size significantly affects segmentation accuracy.

- **Adjusted Window**: Adjusted window techniques date back to the late 20th century when Daley et al. (1998) explored techniques using variograms with false positive filters based on the Getis index to adjust the window size accordingly. They concluded that an algorithm featuring a variable window size surpasses those utilizing a stable one. The adjusted window method was further refined by Popescu et al. (2002); Popescu and Wynne (2004) in the early 2000s, who estimated the adjusted window size by using a linear regression model that modeled the maximum crown radius w.r.t. powers of height. This approach was notably implemented in the study by Picos et al. (2020), where the window size was determined as a function of height with coefficients calculated in previous research by Popescu and Wynne (2004). Similarly, Puliti et al. (2020) applied the LM algorithm to both the Canopy Height Model (CHM) and the Point Cloud Data (PCD), setting window sizes based on the minimum distance between neighboring treetops to include data points from all trees, not just dominant ones, thereby optimizing detection accuracy and reducing segmentation errors.

Comparing the two approaches to determine the optimal window setting remains under-explored. The best choice likely depends on the application context. For example, in a more heterogeneous forest site (e.g., in terms of tree age), using adjusted windows may be more advantageous. Recent benchmarks reveal a trend in favor of adjusted window algorithms, presenting state-of-the-art results in Vauhkonen et al. (2012); Eysn et al. (2015); Wang et al. (2016); Yancho et al. (2019). In their study,

Eysn et al. (2015) examined segmentation algorithms under different forest types, coniferous or deciduous and single- or multilayered, reporting results up to 65% detection rate. They concluded that an LM-based method with a variable-sized window performed best, a result also supported in Daley et al. (1998). Note that, however, their results are not differentiated per canopy class.

An application of adjusted window based on tree height was also implemented in Daley et al. (1998); Popescu et al. (2002); Popescu and Wynne (2004), and later implemented by Picos et al. (2020), an approach that uses the square of the height to model maximum crown radius. Such an approach is limited by the ability of the fitted model to describe the maximum crown radius using only height, a procedure that can lead to inconsistencies when multiple species are involved. Furthermore, in Li et al. (2012), even though the algorithm is LiDAR-based, i.e. applied on the LiDAR dataset and not the CHM, there is again a distinction between the lower and upper canopy.

These works typically determine search windows through empirical reasoning and numerical experiments, exploring various settings. This empirical methodology may inadvertently overlook optimal solutions not included in the testing range and lacks a firm justification. To address, or at least mitigate these shortcomings, we propose refining the adjusted window method by grounding our specification of the window size using the spatial point processes theory. Our investigation proposes an adaptive circular window, varying in size w.r.t. tree height, whose radius is optimized.

To the best of our knowledge, only one study investigated an approach based on the point process theory, namely Ene et al. (2012) with a technique rooted in the Poisson point process. However, Poisson point process assumptions allow points to be arbitrarily close. If we consider points to be tree locations, these points cannot be arbitrarily near each other, mainly because of the trunks, which create 'no-interaction' zones. Thus, our method employs an alternate point process, namely the hard-core point process, which is a particular case of the Strauss process (belonging to the broader family of Gibbs models) and accounts for this 'no-interaction' zone (Moller and Waagepetersen, 2003). Another key difference from the aforementioned study (Ene et al., 2012) is that tree density is modeled via a linear model—whereas in our case, it is a parameter to be inferred from the data.

The following section provides a description of the datasets (simulated, real) used in this study. Subsequently, a concise theoretical overview of the underlying theory applied in this research is presented. A hypothesis test is conducted on the real dataset, for the proposed model, and the segmentation results are presented in Section 3, where the core findings are reported. The final sections engage in a detailed discussion of these findings.

2. Material and Methods

2.1. Simulated dataset

For the examination of the proposed methodology, a synthetic dataset was simulated (Florakis, 2025) mimicking several features present in the real dataset. First, objects representing trees were generated using the Arbaro software (Mueller, 2007), and then the generated trees were placed on a scene (an $80\,\mathrm{m} \times 80\,\mathrm{m}$ area), over which a LiDAR point cloud was generated using the Helios++ software (Winiwarter et al., 2022). The spatial placement of the generated trees is described in Section 3.1. Since the generated point clouds were denser than those of the full real LiDAR dataset (presented in the next section), two distinct thinning strategies were examined. The first one retained points uniformly with a 0.01 probability, while the second one aimed to thin similarly to an Airborne Laser Scanning (ALS) LiDAR acquisition (White et al., 2017), i.e., point penetration is assumed to follow a Beer-Lambert (BL) law of exponential decay (Lambert, 1760; Beer, 1852), retaining points with a probability

$$p(Z) = p_0 + (p_1 - p_0) \cdot \frac{e^{-k_{\mathrm{BL}}(1-Z)} - e^{-k_{\mathrm{BL}}}}{1 - e^{-k_{\mathrm{BL}}}}$$

where $p(Z)$ is the per-point retention probability at the normalized height Z scaled to $[0, 1]$, $p_0 = 0.001$ and $p_1 = 0.04$ are the minimum and maximum retention probabilities respectively, and k_{BL} is the Beer-Lambert coefficient, fixed at a representative value of 0.6. All the other values were chosen manually so that the desired density of $\sim 17\,\mathrm{points}/m^2$ was achieved, close to the observed one ($\sim 17.9\,\mathrm{points}/m^2$).

Different scenes were generated based on species and size composition (Table 1). Species were selected following two approaches: one replicates the observed region, including '*Pinus palustris*', '*Pinus taeda*', '*Liquidambar styraciflua*', and '*Nyssa sylvatica* marshall' (compare with Table 3), the other includes more general species not specific to the area (discussed in the next section), such as 'Black oak', 'European larch', and 'Poplar'. Three size strata were considered as categorical levels: big (25–32 m), medium (15–22 m), and small (5–12 m)—with individual tree heights randomly assigned within each range. Each scene was replicated under different tree densities and spatial configurations, with full details provided in the following sections.

The methodologies examined, as presented in the following sections, were further evaluated under thinning of the original tree positions—that is, under incomplete tree position measurements, a limitation present in the observed dataset that the proposed framework aims to address.

2.2. Real dataset

The available data were downloaded from the National Science Foundation's National Ecological Observatory Network (NEON) collection. Ground truth and LiDAR data come from NEON (2023b) and NEON (2023a). A summary of those can be found

Scene	Species (size classes)
black_oak	Black oak (big, medium),
european_larch	European larch (big, medium, small),
mixforest_gen	Black oak (big, medium); European larch (big, medium, small); Lombardy poplar (big, medium, small),
mixforest	*Liquidambar styraciflua* (big, small); *Nyssa sylvatica* marshall (big, small); *Pinus palustris* (big, medium, small); *Pinus taeda* (big, medium, small),
pinus_mixspecies	*Pinus palustris* (big, medium, small); *Pinus taeda* (big, medium, small),
pinus_palustris_mixsize	*Pinus palustris* (big, medium, small),
pinus_taeda_big	*Pinus taeda* (big),
pinus_taeda_mixsize	*Pinus taeda* (big, medium, small),
poplar	Lombardy poplar (big, medium, small).

Table 1: Experimental scenes and their tree-species/size compositions.

in Table 2. Information on data collection procedures and protocols can be found on Meier (2022) and Krause and Goulden (2022), respectively.

The area of interest extends over 43 km^2 in the Talladega National Forest (TALL), in west-central Alabama (US) (centered on 32.95, -87.39). The typical climate is subtropical, with an average annual temperature of 17.2°C and an average annual precipitation of 1380 mm (PRISM Climate Data Explorer; Alabama State Climate Summary). Soil is characterized primarily by sand, clay, and mudstone from the Typic Hapludults soil subgroup. As a result of this composition, they are dominantly variegated and sandy, loamy and clayey. Some soils within the forest also have hardened bedrock of ironstone (Bowman, 2020; USGS Mineral Resources Data System (MRDS)). The forest, resembling the Upper Coastal Plains in vegetation but with Appalachian Plateau-like topography, is predominantly coniferous. Its canopy is chiefly composed of longleaf and loblolly pines, underlain by various oak species. Notably, over 40% of the area is characterized by hardwoods and wetlands despite its reputation for longleaf pine forests (Hatchner, 2017; National Forests in Alabama). Tree data collection campaigns were conducted during the period $2015 - 2021$. For each tree, only the last chronological observation was considered. In total 376 observations were included in the considered dataset. Table 3 summarises some of the main tree attributes (Height, Maximum Crown Diameter (D_∞), Perpendicular Crown Diameter ($D_\infty^\perp$), i.e. in the direction perpendicular to the maximal one, in the horizontal plane, Diameter at Breast Height (DBH), measured at 1.3 m, w.r.t. the main species found in the area, and canopy class[1]. We consider three classes: (i) 'Full sun' trees ($n = 51$), characterized by crowns fully lit from above and partially from the

[1]Within the dataset, canopy class information is captured under the variable canopyPosition.

sides, standing taller than average without neighboring plants affecting their crown shape; (ii) 'Partially shaded' trees ($n = 176$), receiving full overhead light but minimal side sunlight, of average height, and slightly crowded; and (iii) 'Mostly shaded' trees ($n = 67$), which get little direct overhead light and no side light. The average tree heights are 23.5, 17.8, and 11.7 m w.r.t. canopy class. 'Fully shaded' trees' class had less than 9 observations and was excluded from the analysis. Out of the 376 total observations, 73 lacked canopy class, and 31 belonged to species with fewer than 10 measured trees.

The initial cloud density of the LiDAR data is estimated as 17.9 points/m^2 over an area of 43 km^2. Height normalization of the point cloud is achieved through spatial interpolation using a k-nearest neighbor approach with an inverse-distance weighting, for $k = 10$ the number of k-nearest neighbors and $p = 2$ the power for the inverse distance weighting. The maximum radius for the search was chosen as 50 m. After the normalization, points with a normalized height below 0 and above 100 were filtered out to avoid outliers. Finally, any duplicate points were also cleared. All segmentation algorithms discussed in the sequel were applied to the first return points, also classified as 'high vegetation' by the LiDAR dataset.

After the procedures presented above, the canopy height model (CHM) was derived by projecting maximum height (z) values over 0.5 m $\times$ 0.5 m pixels, a value determined after manual inspection, as one that left less than $1.16 \cdot 10^{-9}\%$ of empty pixels in the final CHM. For smoothing any empty pixels left, a *pitfree* algorithm was utilized (Khosravipour et al., 2014), for the range of heights $0, 3, \ldots, 24$ m. The sub-circle parameter of the algorithm was set to 0.4 m.

Table 2: Summary of NEON datasets.

Dataset ID	**Dataset Name**	**DOI**	**Site ID**	**Years**
DP1.10098.001	Veg.-Structure	10.48443/73ZN-K414	TALL	2015-2021
DP1.30003.001	Discrete-LiDAR	10.48443/XXBY-5A18	TALL	2021

Following Wang et al. (2016); Yancho et al. (2019); Wang et al. (2019), segmentation was evaluated in three nested test conditions, gradually incorporating all canopy classes. Initially, only 'Full sun' trees were considered, and then 'Partially shaded' and 'Mostly shaded' trees were successively added to the testing set. The rationale behind these three test conditions was to examine the algorithm's accuracy in a favorable scenario, where trees are easily detected in the LiDAR dataset, and in some not-so-favorable scenarios, i.e., where trees are not so easily distinguished from their neighbors.

Table 3: Summary statistics w.r.t. (**Top**) Species and (**Bottom**) Canopy Position: the estimated average values are reported along with the standard deviation (values in the parenthesis) for the tree attributes: Height, Maximum Crown Diameter (D_∞), its perpendicular diameter ($D_\infty^\perp$), and Diameter at Breast Height (DBH), with units mentioned in brackets. Below to each species or canopy class label, the number of trees measured is displayed. If data are missing for certain variables, the available data count is provided in the appropriate column beside the standard deviation. Only species/canopy classes with more than 10 observations are reported. Data reported in this table come from campaigns conducted during the period $2015 - 2021$.

Species	Height (m)	DBH (cm)	D_∞ (m)	$D_\infty^\perp$ (m)
Acer	13.7	20.6	9.4	7.5
(n = 17)	(6.0)	(12.8)	(2.7, n = 14)	(2.5, n = 14)
Carya	15.9	18.3	6.2	4.7
(n = 17)	(4.4)	(4.6)	(1.7)	(1.2)
Liquidambar	13.3	18.6	7.8	5.2
(n = 27)	(6.8)	(10.3)	(5.5, n = 21)	(3.1, n = 18)
Liriodendron	20.4	31.3	9.3	8.0
(n = 14)	(7.1)	(14.8)	(2.8, n = 13)	(2.4, n = 12)
Oxydendrum	10.3	18.5	8.8	5.5
(n = 29)	(4.7, n = 28)	(9.5)	(5.1, n = 22)	(2.3, n = 20)
Pinus	17.8	25.5	6.3	4.9
(n = 181)	(5.8)	(13.5)	(3.2, n = 146)	(2.7, n = 145)
Quercus	18.2	25.4	9.4	7.5
(n = 60)	(11.0, n = 59)	(13.9)	(4.2, n = 53)	(3.5, n = 52)
Other	9.6	15.9	8.1	6.6
(n = 31)	(6.1, n = 30)	(11.9, n = 31)	(2.9, n = 15)	(2.5, n = 15)
Canopy Class				
Full sun	23.5	38.7	10.7	8.7
(n = 51)	(6.1)	(13.4)	(4.1)	(3.8)
Partially shaded	17.8	24.5	6.9	5.5
(n = 176)	(6.9)	(11.4)	(3.0)	(2.4)
Mostly shaded	11.7	15.7	5.9	4.1
(n = 67)	(3.7)	(4.9)	(3.2, n = 65)	(1.8, n = 61)
NA	12.3	17.0	-	-
(n = 73)	(7.2, n = 70)	(12.8)	(-, n = 0)	(-, n = 0)

2.3. Point processes

Spatial data analysis is particularly relevant in forest management. Tree locations or clouds of points formed by the LiDAR measurements can be regarded as instances (realizations) of appropriate random (stochastic) processes. This is exactly the framework of (spatial) *point processes*, which studies properties of spatial point configurations by imposing certain assumptions on the underlying random variables. A *spatial point process* is a mathematical object X, designed to model the random selection of countable subsets (finite or infinite) of a space S. Usually, an additional assumption is imposed, ensuring that on bounded subsets of S, only a finite number of points can be selected (local finiteness). In our application context, the observed points are in a bounded observation window.

To formalize the ideas above and justify our modeling approach, in this section we introduce the necessary mathematical background and establish some notation following principally Moller and Waagepetersen (2003). For a comprehensive exploration of point processes, the reader is also directed to Georgii (2006); Daley and Vere-Jones (2008); Illian et al. (2008); Baddeley et al. (2015); Dereudre (2019).

2.3.1. Poisson point process

The simplest and most popular class of point processes is that of Poisson. The Poisson point process (PPP) serves as a tractable model class for modeling and simulating spatial point patterns with 'no interaction' or 'complete spatial randomness', as well as the building block for more advanced point processes. In order to give a formal definition, we first need to introduce the binomial point process and the notions of intensity measure and function. As usual, the Borel subsets of $\mathbb{R}^2$ are denoted by $\mathcal{B}(\mathbb{R}^2)$.

Counting the number of points of a point process X within a measurable subset of S is crucial in understanding its properties, and the symbol $n(\cdot)$ is used to denote the cardinality function. As usual, the lowercase notation x is reserved for the realization of X (defined in S) and its restrictions to a subset $B \subset S$ by x_B and X_B, respectively.

Definition 1. *Let $S \in \mathcal{B}(\mathbb{R}^2)$, $\mathcal{B}(S)$ be the Borel subsets of S, X be a point process on S and $N(B) := n(X_B)$ be the cardinality of the random set X_B for $B \in \mathcal{B}(S)$. The set function $\mu\colon \mathcal{B}(S) \to [0, +\infty]$ given by*

$$\mu(B) = \mathbb{E}[N(B)], \tag{1}$$

is a measure on $\big(S, \mathcal{B}(S)\big)$ called the intensity measure *of X. If, additionally, there exists a non-negative function $\rho(\cdot)$ for which*

$$\mu(B) = \int_B \rho(\xi)\, d\xi, \qquad B \in \mathcal{B}(S), \tag{2}$$

then $\rho(\cdot)$ is called the intensity function *of X.*

Now, we define a Poisson point process.

Definition 2. *A point process X on S is a Poisson point process with intensity function $\rho(\cdot)$ and intensity measure $\mu(\cdot)$, if for all $B \in \mathcal{B}(S)$ with $0 < \mu(B) < +\infty$:*

(i) $N(B) \sim \mathcal{P}\big(\mu(B)\big)$, the Poisson distribution with mean $\mu(B)$;

(ii) conditional on $N(B) = n \in \mathbb{N}^*$, X_B consists of n points distributed independently with $f(\xi) = \rho(\xi)/\mu(B)$.

If $\rho(\xi)$ is constant, we call the process homogeneous.

In the case of a homogeneous PPP, conditioning on $N(B) = n$ results in n independent points uniformly distributed on B.

2.3.2. Hard-core Point Process (Hc)

The Hc process is a special case of a Gibbs process. The general form of the Gibbs process joint conditional density, given n points, at locations $\xi_1, \ldots, \xi_n$ is given by

$$f_n(\xi_1, \ldots, \xi_n) = \exp\big\{ -\phi_n(\xi_1, \ldots, \xi_n)\big\} / \int_{B^n} \exp\big\{ -\phi_n(\xi'_1, \ldots, \xi'_n)\big\} d\xi'_1 \ldots d\xi'_n, \quad (3)$$

where ϕ_n is called the *potential function*. The potential function corresponding to the Hc process is given by

$$\phi_n(\xi_1, \ldots, \xi_n) = \sum_{i>j} \psi(\xi_i - \xi_j), \quad (4)$$

thus allowing only pairwise interactions modeled by

$$\psi(\xi) = \begin{cases} \infty, & \text{if } \|\xi\| \leq R, \\ 0, & \text{if } \|\xi\| > R, \end{cases} \quad (5)$$

where R is a parameter of the process, the so-called *hardcore radius*. This parameter specifies a prohibitive distance within which points cannot be found, while preserving spatial randomness (like the PPP) of a certain intensity beyond this distance. The Hc process is thus repulsive.

With these specifications, the density of the Hc process, up to a proportionality constant in a bounded window B, is given by

$$f(x_B) \propto \begin{cases} \beta^{n(x_B)} & , \text{if } \|\xi - \xi'\| > R \text{ for all } \xi, \xi' \in x_B,\ \xi \neq \xi', \\ 0 & , \text{if } \|\xi - \xi'\| \leq R \text{ for some } \xi, \xi' \in x_B, \end{cases} \quad (6)$$

where $\theta = (\beta, R) \in \mathbb{R}_+^2$ are the parameters of the model, the constant intensity outside the hardcore radius, denoted as by β instead of ρ for this model, and the hardcore

radius, respectively. Restricting the analysis to a bounded window B ensures that the aforementioned density function is well-defined. However, determining the corresponding normalization constant is a non-trivial task (Moller and Waagepetersen, 2003; Daley and Vere-Jones, 2003).

An intuitive yet inefficient method to simulate a Hc point process involves generating a PPP and eliminating points within the specified distance R (Moller and Waagepetersen, 2003; Illian et al., 2008; Baddeley et al., 2015). A more efficient way is to implement a Metropolis-Hastings algorithm (Baddeley et al., 2015).

2.3.3. Related functions

In this section, we introduce two of the most used point-process functions. To avoid unnecessary complications, we consider homogeneous point processes.

Nearest Neighbor (NN) distance distribution function (G). For a homogeneous point process X, the nearest neighbor (NN) distance distribution function (G) is defined as

$$G(r) := \mathbb{P}\left(\inf_{x \in X \setminus \{u\}} \|x - u\| \leq r \;\middle|\; u \in X \right), \qquad r > 0. \tag{7}$$

Empirical estimation of the NN distribution function can be performed using usual methods, treating the distances of the observed point pattern as observations[2]. A naive empirical estimate of the above quantity may be biased due to boundary effects. For this reason, a *Kaplan-Meier correction* is employed. Henceforth, any reference to the empirical estimator of the NN distribution function will specifically denote the version incorporating this correction. A discussion on the empirical estimation of the NN distribution function can be found in Appendix C.

Ripley's K*-function.* For a homogeneous point process X with intensity $\rho > 0$, the Ripley's K-function (Ripley, 1977) is defined as

$$K(r) := \frac{1}{\rho}\, \mathbb{E}\left[N\big(B(u,r) \setminus \{u\}\big) \;\middle|\; u \in X \right], \qquad r > 0, \tag{8}$$

where $B(u, r)$ represents the open ball centered at u with radius r. Thus the function $K(r)$ determines for each distance r, the mean number of supplementary points (other than the center) in the corresponding ball per intensity unit given that the ball is centered at a point of the process.

Note that since the process is assumed to be homogeneous, both functions above do not depend on the choice of u.

[2]Specifically, the distances to the nearest neighbor of each point of the pattern.

2.3.4. Parameter estimation

Maximum likelihood estimation for the Hc model is usually not feasible since, in most cases, the likelihood function associated with the observed point configuration is intractable (Moller and Waagepetersen, 2003; Baddeley et al., 2015). Nevertheless, parameter estimation can be performed by maximizing the pseudo-likelihood (PL) (Jensen and Moller, 1991; Daley and Vere-Jones, 2003; Baddeley et al., 2015) (see Appendix A). Even though the maximum pseudo-likelihood estimator (MPLE) is biased (Baddeley and Turner, 2014), it is asymptotically unbiased, consistent, and asymptotically normal under appropriate regularity conditions (Jensen and Moller, 1991; Jensen and Künsch, 1994; Mase, 1995).

An edge correction strategy must be implemented to eliminate *edge effect bias* (Baddeley et al., 2015). In Ripley (1988), several edge correction methods are discussed and compared. In small datasets (< 500 observations), similar to our case, methods such as translation, isotropic, or rigid motion (translation + isotropic) are recommended (Baddeley et al., 2015). Although the specific choice of edge correction is reported to have limited consequences in many cases, the isotropic method is preferred to counteract the possible inflated variance at greater distances usually exhibited by the translation correction method (Baddeley et al., 2015).

2.3.5. Train and test sets

Training and test sets, over the observed dataset, were determined with particular care to estimate model parameters and evaluate the model's performance. Since the "canopyPosition" attribute was missing in 73 observations, it was necessary to exclude them from results where stratification w.r.t. the canopy class was relevant. Besides, they would impede investigating the cause/source of some potential segmentation errors (e.g. lowest segmentation scores being expected for mostly shaded trees). For this reason, we included the plots with the most unclassified trees in the train set. Finally, we selected all the plots with more than two 'Full sun' observations as the test dataset. In detail, the test set consisted of 9 plots, with 176 trees, out of which 24% ($n = 43$) were characterized as 'Full sun', 44% ($n = 77$) as 'Partially shaded', 24% ($n = 42$) as 'Mostly shaded', and 5.1% ($n = 9$) as unclassified. The rest were used as the training set, that is 27 plots with 200 trees (32.5% unclassified).

2.3.6. Goodness-of-fit testing

A hypothesis test was conducted to examine the validity of an Hc model. In particular, the null hypothesis (H_0), which we aimed to testing, states that the points representing the tree locations come from an Hc model with any possible $\theta = (\beta, R) \in \mathbb{R}^2_+$, while the alternative hypothesis (H_1) represents any other possible model. To test this hypothesis, the estimated NN distribution function served as the test statistic, combined with a parametric bootstrap method to construct a confidence band—a standard practice in spatial point process analysis (Moller and Waagepetersen, 2003; Baddeley et al., 2015). For completeness, we present the entire algorithm in Algorithm 1, which we now describe.

Since there is no closed formula for the NN distribution function, we opted for Monte Carlo sampling to estimate the theoretical values. The samples for this procedure were generated conditionally on the observed number of points of the original dataset. For the NN distribution function, the conditional generation of points has been illustrated to yield narrower envelopes, resulting in more robust outcomes (Møller and Vihrs, 2022). However, a numerical approximation of the NN distribution function is also feasible (Al-Hourani et al., 2016).

In particular, first, a Hc model was fitted on the training dataset, resulting in an estimated parameter $\theta_0 = (\beta_0, r_0)$, where here r_0 (instead of R_0) corresponds to the estimated hardcore radius (the minimal distance between tree tops) (Step 1 in Algorithm 1). Then, $B = 200$ Hc bootstrap point samples of the same size as the original sample were drawn with $\theta = \theta_0$ (the estimated one), noting that this $B = 200$ should not be confused with the train sample size. For each simulated bootstrap point sample, a *Metropolis-Hastings* algorithm was used, thus obtaining a (parametric) bootstrap sample of empirical NN distribution functions, denoted here by $\{\widehat{G}_b\}_{b=1}^{B}$. The theoretical NN distribution function for the Hardcore process, denoted as $\widehat{G}_{\text{theo}}$, was estimated by simulating another 200 point patterns under H_0, of a size equal to the original sample (Step 8 in Algorithm 1). Let $\widehat{G}$ denote the empirical estimator of the nearest-neighbor distribution function from the initial point sample, then the Hc model is not rejected at a significance level $\frac{1}{B+1} \approx 0.005$ (Moller and Waagepetersen (2003) relation (4.20)), provided that $\widehat{G}$ lies entirely inside the 0.995–confidence band, or equivalently if $\|\widehat{G} - \widehat{G}_{\text{theo}}\|_\infty \leq D_{\max}$, where $D_{\max} = \max_b \|\widehat{G}_b - \widehat{G}_{\text{theo}}\|_\infty$, i.e. the maximum observed value of the Kolmogorov-Smirnov distance between $\widehat{G}_b$ and $\widehat{G}_{\text{theo}}$ for all $b = 1, \ldots, B$.

Each bootstrap point sample was simulated via the Metropolis-Hastings algorithm in a single window of equal area as the union of all plots in the training set. Employing a single window rather than multiple ones was crucial to prevent low acceptance probabilities caused by the sparse and small regions of the individual plots. This procedure can be justified by the *translation invariant* property of the stationary Hc process, conditionally on the same number of points. This step was essential due to the lack of a closed-form solution for the nearest neighbor distribution; even existing formulations, such as those noted by Al-Hourani et al. (2016), require numerical approximations.

This Bootstrap-based hypothesis test is known to be conservative (Moller and Waagepetersen, 2003; Baddeley et al., 2015) and can introduce bias in the subsequent inference. For this reason, we added an extra step to the algorithm, suggested by Dao and Genton (2014). We present a simple version of this methodology and postpone a more detailed discussion for the Appendix B. After each point pattern sampling from the estimated parameters under the null hypothesis, i.e., from the estimated $\theta_0 = (\beta_0, r_0)$, extra point patterns are simulated, w.r.t. the estimated parameters of the new samples, resulting in an equal number of estimated

p-values. The quantiles of these values are used to estimate an *adjusting confidence level* w.r.t. the initial confidence level. The initial Monte Carlo estimation of the p-value is then compared with this *adjusted level*. This test is asymptotically exact, i.e., as the number of simulations grows, the distribution of the estimated p-values converges to the uniform distribution, a result that simulated experiments have further demonstrated in Dao and Genton (2014); Baddeley et al. (2015).

Algorithm 1: Bootstrap goodness-of-fit test for Hc process.

Data: Observed point configuration $x = \{x_1, x_2, \ldots, x_n\}$ from the point process X in a bounded set $S = \bigcup_{i=1}^{m} S_i$, where $S_i \cap S_j = \emptyset$, $i \neq j$, and $\{S_i\}_{i=1}^{m}$ represent the m different available plots.

Result: Bootstrap goodness-of-fit test result.

1 *Estimate:* the parameter θ_0 out of the sample X (maximize the Pseudolikelihood);

2 *Initialize:* Set W a square of side $\sqrt{\lambda(S)}$. Set number of bootstrap samples B and initialize two objects, one for bootstrap estimates $\widehat{G}_{\text{bootstrap}}$, and one for the theoretical values of $\widehat{G}_{\text{theo}}$;

3 **for** $b \leftarrow 1$ **to** B **do**

4 Generate sample X_b^* of a Hc with parameter θ_0, over W, of equal size n as X;

5 Calculate the empirical NN distribution function $\widehat{G}_b$ based on the sample X_b^* ;

6 **end**

7 *(Theoretical NN distribution):* Generate M samples X^* of size n from Hc with parameter θ_0. Estimate the theoretical G with

$$\widehat{G}_{\text{theo}} = \frac{1}{M} \sum_{i=1}^{M} \widehat{G}_i$$

and calculate the empirical NN distribution function $\widehat{G}$ using a Kaplan-Meier correction (C.3).

8 *(Confidence bounds):* Calculate the values

$$D_{\max} = \max_b \|\widehat{G}_{\text{theo}} - \widehat{G}_b\|_\infty$$

Output: **Reject** if for any r

$$\widehat{G}(r) > \widehat{G}_{\text{theo}}(r) + D_{\max}$$
$$\text{or } \widehat{G}(r) < \widehat{G}_{\text{theo}}(r) - D_{\max}.$$

Retain otherwise.

9 *(Optional Dao-Genton correction):* Repeat steps 7,8 for (extra) samples X_b^* $(b = 1, \ldots, B')$. A detailed discussion can be found in Appendix B.

2.4. Adaptive window approach

In this section, we describe our main methodological contribution to the problem of estimating an adaptive window width w for the LM algorithm. Given that the Hc model is retained with the hypothesis testing described in the previous section (see further Section 3), the estimated parameter θ_0 uniquely determines a distribution upon which the adjustment of w could be guided with the help of an appropriate statistic. For this purpose, many choices exist, and a convenient and intuitively appealing statistic is the Ripley's K-function that we denote here by K_{θ_0} to declare the parameter under which the expectation in (8) is computed. In fact, K_{θ_0} allows us to determine the choice of w in such a way that the expected number of trees (other than the center) in the disk of radius w centered at a tree is 1. In general, if we denote by $\mu > 0$ the aforementioned expected number, then determining an appropriate window size $\widehat{r}$ depends on solving (8) by inverting the Ripley's K-function. We put

$$\widetilde{r}(\mu) = K_{\theta_0}^{-1}(\mu/\lambda_0), \tag{9}$$

where λ_0 is the intensity of the process under $\theta_0 = (\beta_0, r_0)$. For an explicit form of this intensity, interested readers are referred to Al-Hourani et al. (2016); Chiu et al. (2013) and in practice (9) is solved by numerical inversion.

The resulting solution depends clearly on the choice of μ. As noted above, the choice $\mu = 1$ is intuitive, as it corresponds to the expected distance to the nearest neighbor — yielding roughly one tree per ball — and would likely be effective under a complete data scenario where tree positions are fully known. However, typically, plot measurements do not include all the trees, and this is also true for our real dataset. The resulting limitation poses the challenge of determining an appropriate choice of μ in the right framework. In fact, since only a subset of the original points is available, this could be thought of as a *thinning* of the initial process, which is properly defined as retaining initial points with some probability p (Moller and Waagepetersen, 2003). Applying a 'reasonable' choice to the thinned point process could introduce bias. In the next paragraph, we illustrate this fact with the NN distribution function, which is easier to handle. We start by drawing a connection between G and K functions.

Because G and K-function are injections, for the 'optimal' choice of $\mu = 1$ there exists an $\alpha^\star$ such that

$$G_{\theta_0}^{-1}(\alpha^\star) = K_{\theta_0}^{-1}(1/\lambda_0) = \widetilde{r}(1). \tag{10}$$

Let pX_{hc} denote a thinning realization of X_{hc}, and $G_{pX_{hc}}$ and $G_{X_{hc}}$ denote their NN distance distribution function, respectively. Since we retain only some of the initial points, it is evident that

$$G_{pX_{hc}} \leq G_{X_{hc}}, \tag{11}$$

Applying the previous methodology to the complete dataset, we can choose an optimal

$r_{\alpha^\star}$ such that

$$r_{\alpha^\star} = G^{-1}_{X_{hc}}(\alpha^\star). \tag{12}$$

A naive application of this optimal choice ($\alpha^\star$) to the thinned process results in

$$\widehat{r}_{\alpha^\star} = G^{-1}_{pX_{hc}}(\alpha^\star). \tag{13}$$

In both cases,

$$G_{pX_{hc}}(\widehat{r}_{\alpha^\star}) = G_{X_{hc}}(r_{\alpha^\star}) = \alpha^\star, \tag{14}$$

which, combined with (11), results in

$$\widehat{r}_{\alpha^\star} \geq r_{\alpha^\star}. \tag{15}$$

This naive application yielded a positively biased estimate of $\alpha^\star$. This holds since G is increasing in any case and justifies our reasoning.

Some final remarks are required. First, the Hc process was chosen for convenience and alignment with the text, but similar reasoning leads to the same result when thinning is applied to a general stationary process. Secondly, when $\alpha \to 0$, $\widehat{r}_\alpha$ also converges to 0 and not $r_{\alpha^\star}$.

If the probability of thinning (p) was known, a reasonable (and optimal) choice for α would have been $p\,\alpha^\star$. Since this probability is unknown, we can examine all the values of $\alpha \in [0, 1]$. To avoid computational burdens, the search region can be narrowed to $[0, 0.64]$, with 0.64 an upper bound for $\alpha^\star$, derived from the PPP. A detailed discussion on this topic is provided in Appendix D.

2.4.1. Adapting to height

To obtain an adaptive window over different height ranges, we define a series of distinct heights such that $0 < h_1 < h_2 < \cdots < h_n < \infty$. Correspondingly, we establish overlapping height classes in the form $[h_1, \infty), [h_2, \infty), \ldots, [h_n, \infty)$. The reasoning behind the overlapping classes is that for lower tree tops, it is essential to account for all trees that might conceal or intersect with their tops. Conversely, lower tree tops are less relevant in the study of taller trees, allowing a robust examination of the upper canopy independently.

Applying the estimation procedure from Section 2.3.4 to the training dataset for these classes yields estimates for each class, resulting in a piecewise constant window width function w.r.t. height over the intervals $\{(h_i, h_{i+1})\}_{i=1,\ldots,n}$. Finally, a 6 knots cubic spline (Lai and Schumaker, 2007) was used on the estimated piece-wise results for smoothing purposes.

It is important to note that the accuracy of this estimation is closely linked to the length of the intervals between successive heights, $(h_{i+1} - h_i)_{i=1,\ldots,n-1}$, as long as the interval length is not too big, which could result in big jumps, due to the

exclusion of many observations. However, smaller interval values do not significantly enhance accuracy, since they result in many constant estimates until the next jump. In our case, a constant step of 0.5 m is used between successive heights, a manually tuned value that results in small regular jumps, starting from the initial height of 2 m, a height generally suggested in the LiDAR bibliography as the lower height for considering trees (White et al., 2013, 2017), and up to a maximum height of $h_n = 20$ m beyond which there are not enough data left.

2.5. Estimating errors

Window size estimators.. To estimate the error of various estimators derived from quantiles (as detailed in Equation (9)), the jackknife methodology (Quenouille, 1949) was employed over the m training plots, with $m = 27$. In this context, $\widehat{r}_a$ represents the estimator generated by including all the plots, while $\widehat{r}_a^{-i}$ symbolizes the same estimator but with the exclusion of the i-th plot. The formula for the jackknife bias (b_{jack}) among m plots is given by

$$b_{\text{jack}}(\widehat{r}_a) = (m-1)\left(\frac{1}{m}\sum_{i=1}^{m}\widehat{r}_a^{-i} - \widehat{r}_a\right), \tag{16}$$

while for the jackknife standard deviation ($\widetilde{s}_{\text{jack}}$)

$$s_{\text{jack}}(\widehat{r}_a) = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(\widetilde{r}_a(i) - \frac{1}{m}\sum_{j=1}^{m}\widetilde{r}_a(j)\right)^2}, \tag{17}$$

where $\widetilde{r}_a(i) = m\,\widehat{r}_a - (m-1)\,\widehat{r}_a^{-i}$ the so called pseudo-values (Wasserman, 2006). The values of both estimators are reported in meters (m). When $\widehat{r}_a$ are compared w.r.t. b_{jack}, the absolute value of this estimator is used.

Segmentation accuracy.. The segmentation procedure results in clustering the 3D point cloud points into different classes that, hopefully, represent trees. By projecting these points into the 2D plane and considering the convex set that the projections generate, we end up with 2-dimensional polygons. These polygons will be used to estimate the accuracy of the segmentation.

Due to the fact that not all trees were measured, not all standard confusion metrics can be computed. We denote N_{poly}, the total number of polygons containing at least a tree, and N_{trees}, the total number of trees in the test dataset. Segmentation performance was measured in two ways:

- Percentage of **correctly identified polygons** (precision):

$$\frac{1}{N_{\text{poly}}}\sum_{i=1}^{N_{\text{poly}}}\mathbb{1}_{\{\text{exactly one tree in the polygon } i\}}, \tag{18}$$

- Percentage of **correctly identified trees** (sensitivity):

$$\frac{1}{N_{\text{trees}}} \sum_{i=1}^{N_{\text{trees}}} \mathbb{1}_{\{\text{tree } i \text{ is uniquely contained within a polygon}\}}, \tag{19}$$

where $\mathbb{1}_A$ is the indicator function of the set A, which takes the value 1 if the condition of the set is satisfied, 0 otherwise. We then calculate F_1-score as

$$F_1 = 2 \cdot \frac{\text{sensitivity} \cdot \text{precision}}{\text{sensitivity} + \text{precision}}. \tag{20}$$

Only polygons with at least one authentic tree position were considered in the described analyses. Indeed, due to incomplete tree measurements within each plot (Meier, 2022), distinguishing between truly empty areas (true negatives) and unmeasured areas was unfeasible.

To enhance the F_1-score reliability, we incorporated a bootstrap re-sampling method (B=100), w.r.t. different test plots, for accuracy assessment and reported the statistic's bootstrap estimate.

2.6. Benchmark approaches

Ground truth data was split into training and test datasets to test these accuracies as described in Section 2.3.5. The parameters were estimated on the training set, and the accuracies were calculated for the test set. The estimated adaptive window, as detailed in Section 2.4, was included in a Dalponte2016 implementation (Dalponte and Coomes, 2016). For comparison, we used a Watershed algorithm, similar to Jaakkola et al. (2017), which is one of the most used algorithms in our (LiDAR) context (Chehreh et al., 2023), the Li2012 algorithm (Li et al., 2012), two approaches similar to Popescu and Wynne (2004), and three instances 3 cases of the Dalponte2016 algorithm (Dalponte and Coomes, 2016) with fixed circular window sizes of $1, 1.5$, and 2.5 m radius.

2.6.1. Watershed algorithm

The Watershed algorithm is an image processing technique used for segmentation in which an image is processed as a topographic map and segmented into distinct regions by simulating the rise in water levels from various starting points and creating boundaries at points where these waters would meet. The tolerance for this algorithm was chosen as 0.01 m, optimized through a grid search over the values $0.001, 0.01, 0.05, 0.1, 0.25, 0.5$, and 1 m. The window size was set to 1 m radius by a grid search over the values $0.75, 1, 1.5, 2, 2.5, 3$.

2.6.2. Li2012 algorithm

The method developed by Li et al. (2012) employs a 'top-down' approach for segmenting data, initiating segmentation from the 'global maxima' (GM) of an unseg-

mented dataset. This point serves as the starting point for each subsequent tree, with tree construction guided by specific rules for adding points. These rules consider the elevation of a point and its status as a 'local maxima' (LM) within a two-dimensional search zone. Due to the criteria for organizing points, this technique is particularly suited for analyzing coniferous forests, where the apexes of trees often exhibit pronounced conical shapes. This segmentation routine by Li et al. (2012) has proven to be highly effective across various scenarios. Their research successfully identified 86.05% of trees within their study areas, amounting to 327 out of 380 trees. The parameters $d_1 = 1.5\,\text{m}$, $d_2 = 2\,\text{m}$, and $Z_u = 15\,\text{m}$ were used, suggested by the original publication, as default values. The Li2012 algorithm is mainly used as a benchmark of a no-tuned algorithm.

2.6.3. Popescu approaches

We further included in the comparison two different cases of the approach used in Popescu and Wynne (2004); Picos et al. (2020). In this approach, a linear model is fitted with the maximum crown radius as the dependent variable and the height as the explanatory variable. We examined two cases, one with the parameters suggested in the aforementioned publications (*Popescu*) and the other by re-estimating the parameters (*Popescu ft*).

2.7. Implementation

The coding was done in R language (version 4.2.3 (2023-03-15 ucrt))[3]. Most of the data processing and plotting was performed utilizing the *tidyverse* package (Wickham et al., 2019). Algorithms for processing LiDAR data were implemented using the *lidR* package (Roussel et al., 2020). Splines were implemented using the *mgcv* package (Wood and Wood, 2015). Spatial models were handled using the *spatstat* package (Baddeley et al., 2015).

3. Results

3.1. Simulation study

In this section, the results of the simulation study, described in Section 2.1, are presented. The $80\,\text{m} \times 80\,\text{m}$ scenes were replicated four times under three different spatial configurations, resulting in twelve total replications. For the first two, the Hc process model 6 was used with parameters $\beta = 0.01, R = 0.5$ (low intensity) and $\beta = 0.02, R = 0.5$ (high intensity), respectively. For the third, a uniformly random placement of 120 tree objects was made on a grid, with vertices placed every 0.5 meters (grid). Per-point retention probabilities for the tree position thinning analysis ranged from 0.3 to 1, with the lower bound manually set to prevent inversion issues in

[3]Code for he implementation on the simulated dataset is available in `https://gitlab-research.centralesupelec.fr/konstantinos.florakis/tuning_dalponte2016` (commit 6a9f3b2a).

window size computation, observed for lower values. We further note that for three cases ('poplar – grid', 'pinus_taeda_big – high intensity', and 'pinus_taeda_mixsize – high intensity'), computational issues arose during scene generation, and these cases were therefore excluded from the analysis.

Along with the proposed methodology, all the benchmark approaches were tested too, with some slight modifications. For methods employing a fixed window (e.g., watershed, Dalponte2016 stable window ('Dalp_stable')), the window radius was set to the value $\sqrt{(\pi\beta)^{-1}}$, where β is the true parameter of the corresponding Hc process—yielding approximately one point per ball and thus providing an oracle advantage to these methods. In the 'grid' spatial arrangement case, the radius was instead set to $\sqrt{(0.02\,\pi)^{-1}} \approx 4$, based on an estimated tree density of 120 trees per 80^2 m^2 (i.e., ≈ 0.02). An additional Dalponte2016 stable window variant ('Dalp_stable_mini') with a fixed window diameter of 1 m was included to illustrate over-segmentation effects, which are discussed later in the text. For the comparison of the methodologies, we used the F_1-score and the Mean Absolute Percentage Error (MAPE) on the predicted number of trees $\widehat{n}_{\text{trees}}$, defined as

$$\text{MAPE} = \left| \frac{\widehat{n}_{\text{trees}} - n_{\text{trees}}}{n_{\text{trees}}} \right|,$$

where n_{trees} the true number of simulated trees. We note that the F_1-score calculation here is exact, since all the locations are known.

In Figure 1 the point cloud thinning strategies are compared, i.e., ALS and uniform thinning of the point cloud points, as discussed in Section 2.1. Using a Wilcoxon signed-rank test on the paired samples, we do not reject the null hypothesis that the median pair-wise difference is zero ($p \approx 0.25$); i.e., the two strategies indicate equivalent performance.

Figure 2 presents the comparison of the different methodologies in the different settings of species composition presented in Table 1. For the 'Hard-core' method, only the optimal choice of quantile (minimizing MAPE) is displayed. Additionally, methodologies with MAPE $> 100\%$ are omitted from the corresponding plot. The proposed methodology ('Hard-core') achieves the lowest MAPE for 81% of the total cases ($n = 324$), while obtaining the second-best score in 16% of them. There are only four cases where the methodology scores lower than the second-best: three under the combination ('pinus_taeda_big', 'grid') and one more for the combination ('mixforest_gen', 'low intensity'). Regarding the F_1-score, the method ranks first in 37% of the cases and second in 46% of them.

Finally, Figure 3 shows the optimal quantile (x-axis) that minimizes MAPE across different retention probabilities (y-axis) used for thinning the original simulated tree positions. Point shapes denote spatial configurations: circles for grid, squares for low intensity, and triangles for high intensity. An approximately linear trend is observed in most cases, consistent with the analysis in Section 2.4, except for 'pinus_taeda_big', where the optimal choice shifts to the upper bound of the considered range, the 0.8-

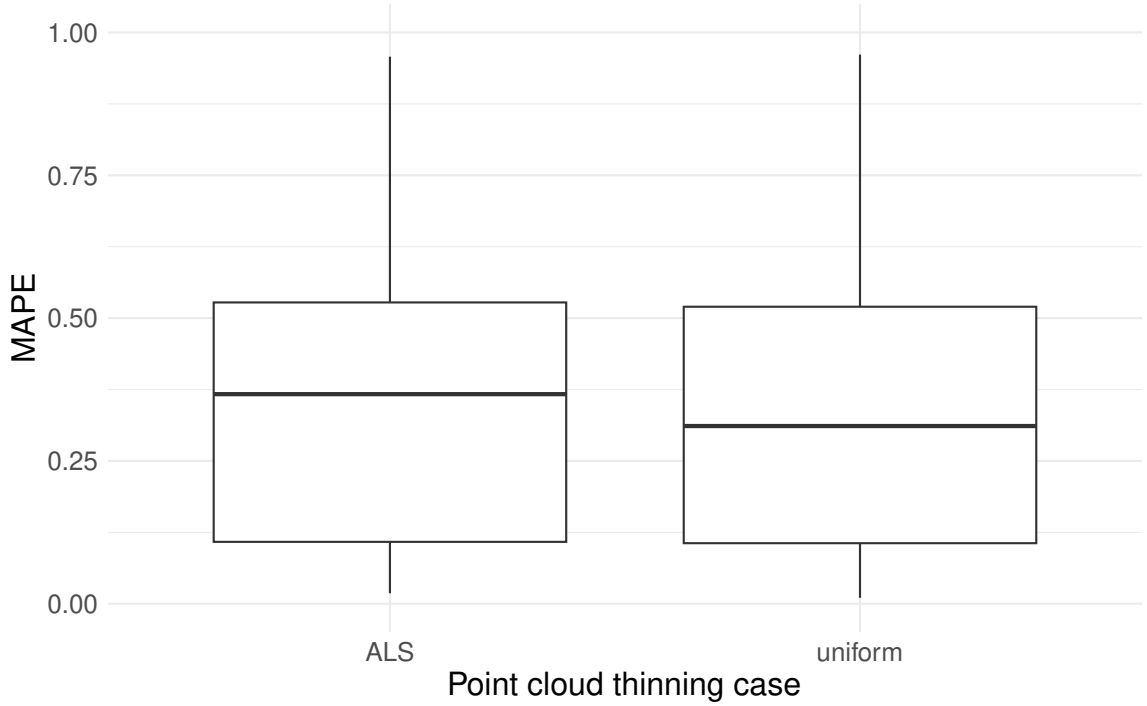


Figure 1: Comparison over different point cloud thinning approaches as discussed in Section 2.1.

quantile.

3.2. Application

Figure 4 displays the hypothesis test results through confidence bands visualization, as presented in Section 2.3.6. The observed nearest neighbor distribution is observed inside the confidence bands, indicating failure to reject the null hypothesis. The estimated values for the model's parameters are included as an annotation. The complete set of parameters across the various classes can be found in Table E.4. In Figure 5, estimations of radii for circular windows using various quantile estimators as outlined in Section 2.4 are presented. Each line shows the radius size in meters, with quantiles beginning at 0 and incrementing by 0.05, as indicated by the labels. The lines do not intersect, an expected result since they are based on quantiles. For most lines, the change rate is observed as small until the height of 12.5 m. A spike observed around the height of 16.25 m is attributed to a sudden change in the number of observations.

The jackknife estimators (Equations (16), (17)) are computed on the training dataset and presented in Figure 6. The x-axis displays different values of α that represent the window size estimators ($\widehat{r}_\alpha$). The dashed lines denote the minimum in each case, $b_{\text{jack}}(\widehat{r}_{0.275}) = 0.06$ (4 % rel. error) and $s_{\text{jack}}(\widehat{r}_{0.275}) = 0.11$ (7 % rel. error), while the vertical dotted lines point out the value of $\alpha = 0.275$, where the minimum is achieved. For the bias, the minimum is calculated w.r.t. the absolute

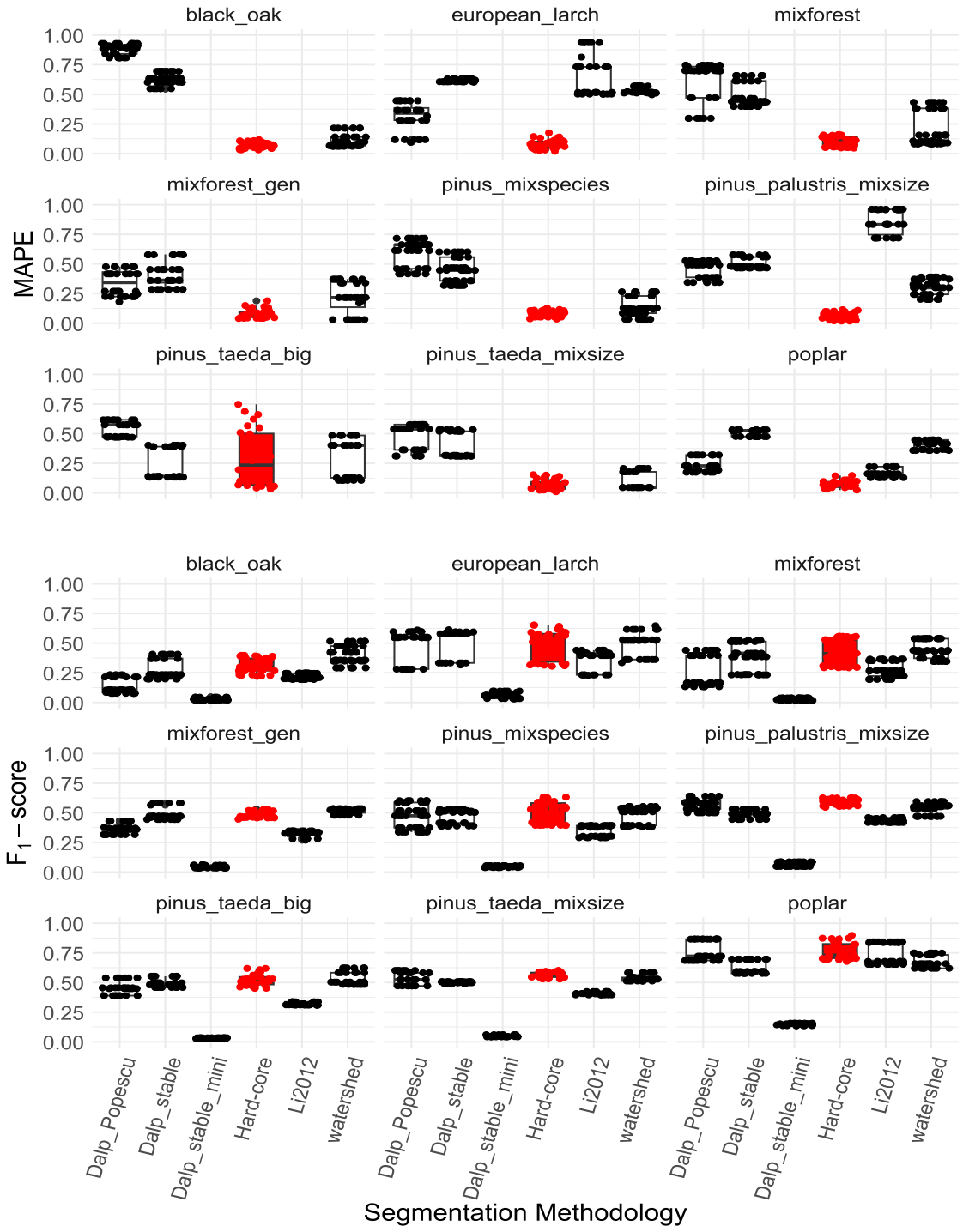


Figure 2: Performance of segmentation methods over varying species compositions (detailed in graph subtitles; refer also to Table 1). MAPE (top) and F_1-score (bottom) results are shown, with methodologies aligned on the common x-axis. Red colouring indicates the results of the proposed methodology ('Hard-core').

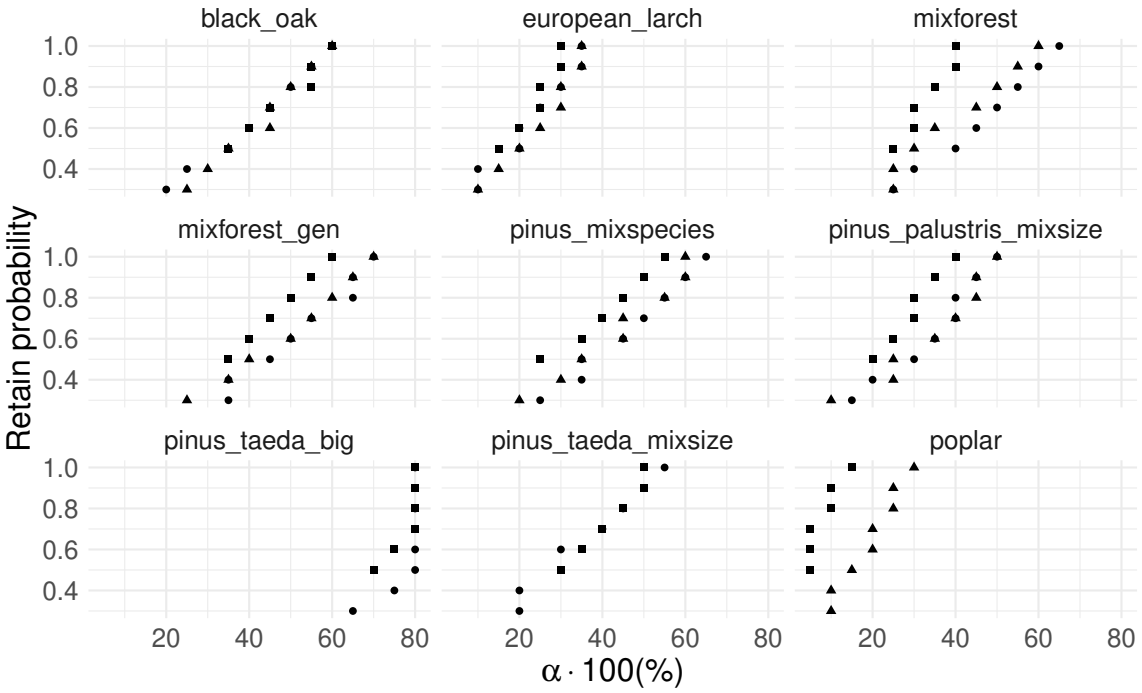


Figure 3: Comparison of the optimal quantile choice w.r.t. MAPE (x-axis) for each retention probability (y-axis) used for thinning the original simulated tree positions. Different spatial configuration cases are indicated by distinct point shapes (circles: 'grid', squares: 'low intensity', triangles: 'high intensity'). In some cases, overlap is observed.

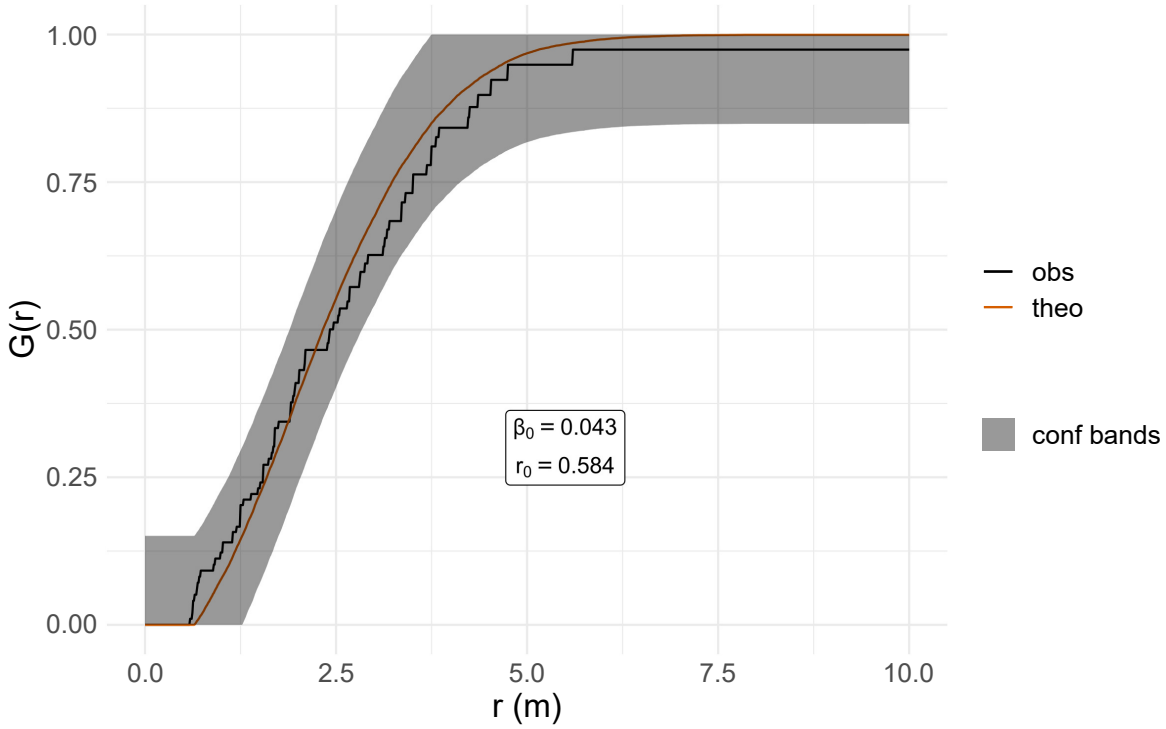


Figure 4: Dao-Genton envelopes (Dao and Genton, 2014) for the Hypothesis test as outlined in Section 2.3.6. The red line is the theoretical value of the nearest neighbor distribution of the Hc process under the null hypothesis, the black line is the observed one, and the grey area indicates the envelopes under H_0. The estimated parameters, under H_0, can be found inside the annotation box.

value of the estimator. The shaded area indicates α values in $[0.075, 0.64]$. Values of α smaller than 0.075 were excluded from the analysis since the resulting circular window has a radius less than 0.5 m, as can be observed in Figure 5, the pixel side of the initial CHM. We further report that the relative jackknife standard deviation of the estimated parameters is less than 10 % for the intensity (β_0) and less than 5 % for the *hardcore distance* (r_0). Figure 7 presents the F_1-score results, as described

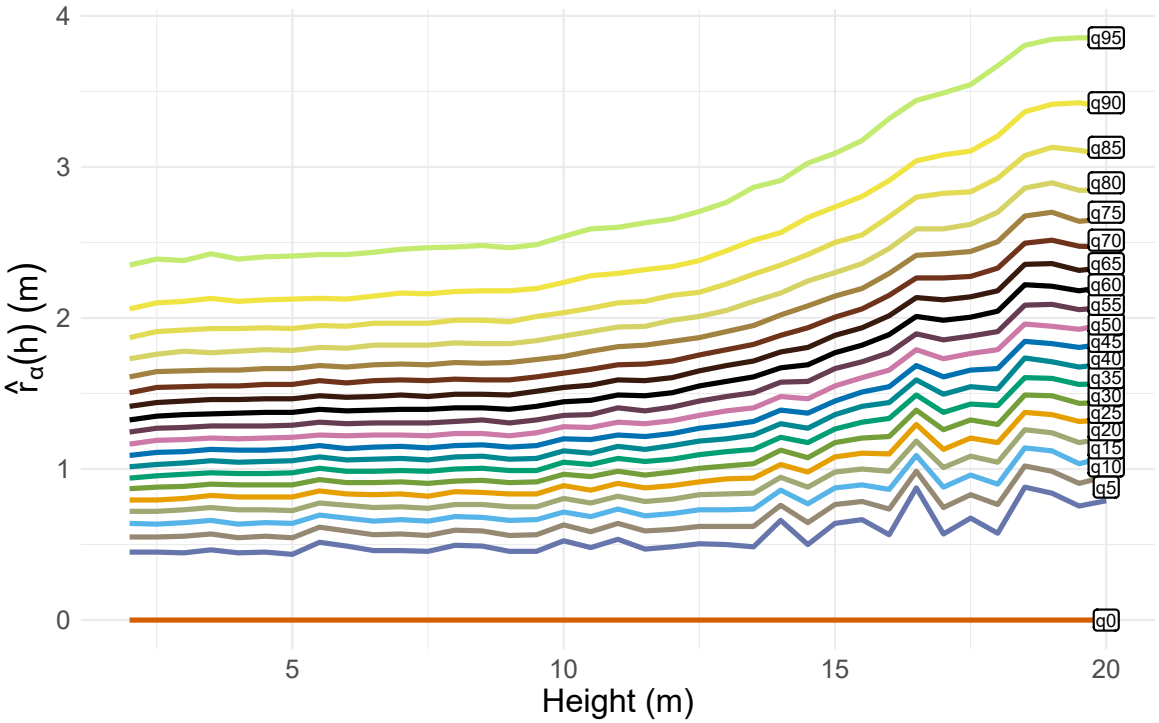


Figure 5: Circular window radius ($\widehat{r}_\alpha$) estimates for different quantile estimators, as a function of height (h), according to the methodology detailed in Section 2.4. Each line represents the window size radius in meters (m). Quantile probabilities start from 0 (bottom line) and increase by 0.05, as the annotations describe.

in Section 2.5. The curved line represents the different estimators w.r.t. α choice. Comparatively, dark blue illustrates the Watershed algorithm's performance, and light blue corresponds to the Li2012 algorithm, while light green, black, and brown lines represent the Dalponte algorithm's application with stable windows 1, 1.5, and 2.5 m radius. Yellow and dark green lines follow methodologies akin to those in Popescu and Wynne (2004); Picos et al. (2020) using pre-estimated parameters and with re-estimation, respectively. We further report that precision and sensitivity, of the proposed methodology, respectively range in (0.90, 0.99) and (0.84, 0.95) for 'Full sun', (0.60, 0.77) and (0.39, 0.61) for 'Partially shaded', and (0.48, 0.61) and (0.27, 0.43) for 'Mostly shaded'.

Segmentation results can be observed in Figure 8. The selection of the specific plot ('TALL_007', as coded in the dataset NEON (2023b)), was made to allow for

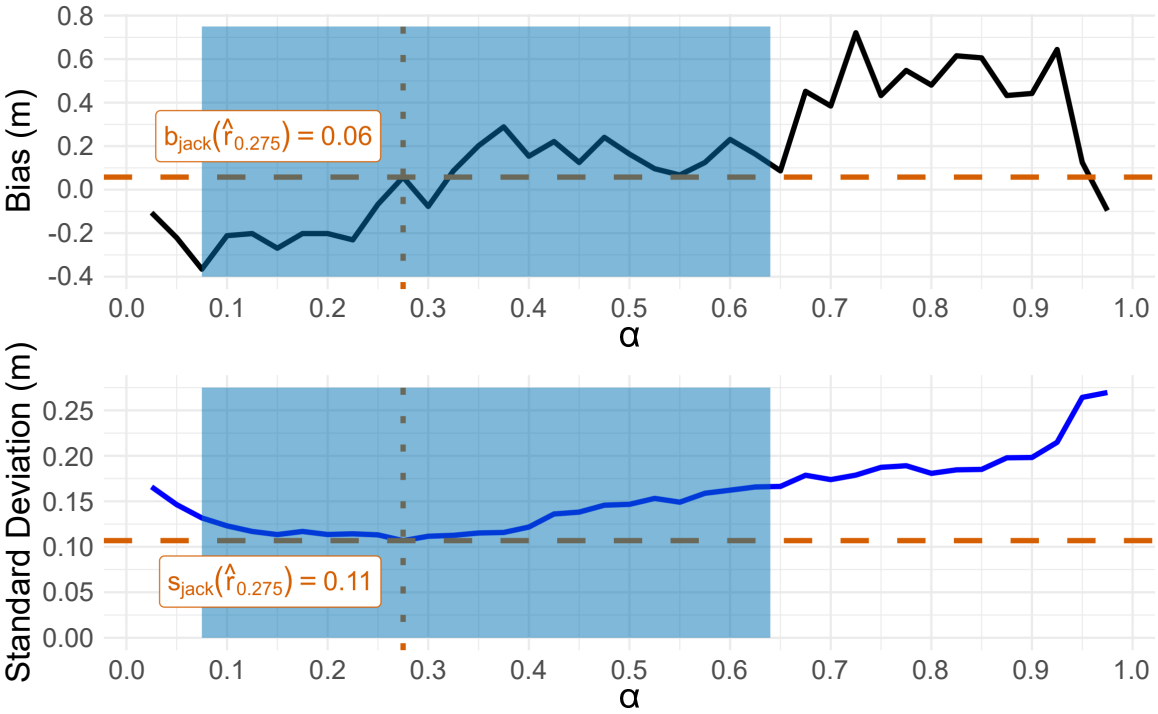


Figure 6: Different quantile estimators of window function $\widehat{r}_\alpha$ (x-axis) compared under the estimated jackknife bias (top) and standard deviation (bottom) (Section 2.5). Values of α range between $[0.025, 0.975]$ with a 0.025 step. The shaded area displays the α values in $[0.075, 0.64]$. The horizontal dashed lines indicate the achieved minimum w.r.t. the absolute jackknife bias and jackknife standard deviation, respectively, achieved for $\alpha = 0.275$.

the demonstration of many cases. Grey areas represent the segmented crown, red points represent the documented positions of trees characterized as 'Full sun', yellow points represent the 'Partially shaded', and blue points denote the 'Mostly shaded' ones. Panels (A) to (F) correspond to different quantile-derived window functions $\widehat{r}_\alpha$: 0.075, 0.125, 0.175, 0.225, 0.275, and 0.3, respectively. Starting from (A), we can observe a finer segmentation, which gradually weakens as we consider higher quantile estimators. Clear misclassifications can be observed in the middle of the images, where some 'Partially shaded' trees are found in the same polygon in all the segmentation cases. The two northwest points are misclassified in the last two cases (E-F) but are correctly classified in the other segmentations. Another interesting observation regarding the two middle-west points is that they are contained in the same polygon in (A) but are perfectly segmented for case (B), indicating that smaller values of window size can affect the total segmentation arrangement. A remark should be made for the middle region. There, two segmented areas are present for the segmentations (A-C), while for the cases (E-F), no apparent change is visible, except a mild grouping at the bottom right corner. A single segmented area is present only for case (D). A visual inspection of the LiDAR dataset favors the latter case and reveals a single small tree.

4. Discussion

4.1. Summary of the main findings

The primary objective of this study was to refine the LM method, to assess its performance across different simulated scenarios, and validate it on real data under varying canopy class conditions. Simulation results showed equal or better performance than existing approaches. The results presented in Figure 2 indicate the robustness of the method under different simulated forest types, while a fine adaptability of the method is observed w.r.t. the thinning of the 'observed' tree positions as presented in Figure 3. Findings on real data are consistent with those of Wang et al. (2016); Yancho et al. (2019); Wang et al. (2019) regarding the lower accuracies for suppressed trees, as can be observed in Figures 7 and 8. The Hard-core model assumption yielded a good fit as observed in Figure 4. The accuracy results indicate that the LM-based methods outperform the other tested methodologies and that stable windows can present results similar to those of the adjusted one, as can be observed in Figure 7, a result not directly supported by the simulation study. We attribute the latter result to the lack of complete data on tree positions. This limitation impacts the accuracy of false positive estimates and blurs the comprehensive understanding of the findings. Notwithstanding these constraints, the applied methodology showcased promising capabilities, evidenced by its smooth adaptation across different stable windows accuracies and retain probabilities as demonstrated in Figures 3 and 7. We think that for values of $\alpha \leq 0.2$, in Figure 7, where the results are equivalent to the stable circular window of radius 1 m, over-segmentation is present,

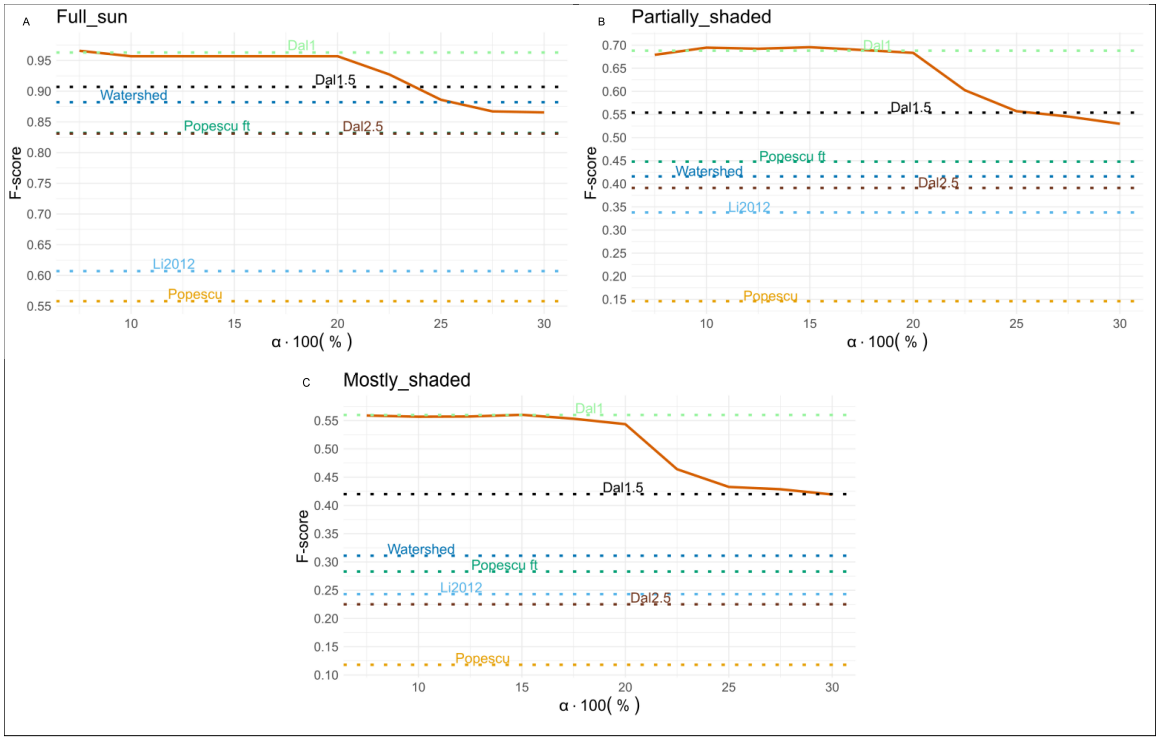
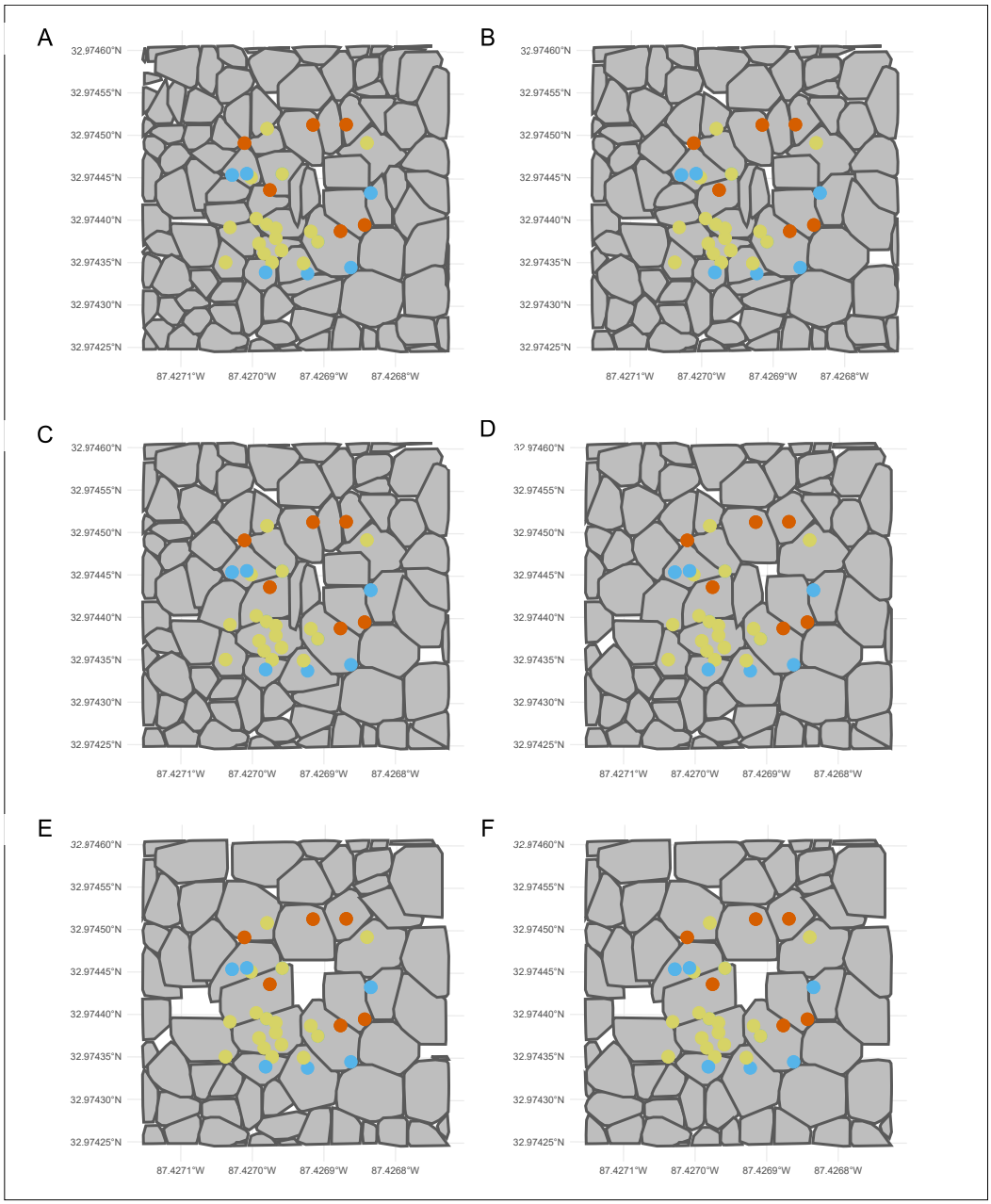


Figure 7: Results of Segmentation accuracy: (A) Only 'Full sun' (B) with the inclusion of 'Partially shaded' (Section 2.5) (C) with the further inclusion of 'Mostly shaded'. Different quantile estimators of the window function $\widehat{r}_\alpha$, as presented in Section 2.4 are compared under F_1-score criterion (20). The red line represents the performance of those estimators. The dark blue line represents the Watershed algorithm's result, and the light blue represents the result of the Li2012 algorithm. The yellow line represents an adjusted window similar to the one followed in Popescu and Wynne (2004); Picos et al. (2020), while dark green represents a similar methodology with a re-estimation of the parameters. Results from the Dalponte algorithm using stable window radii of 1, 1.5, and 2.5 m are shown in light green, black, and brown, and are denoted as 'Dal1', 'Dal1.5', and 'Dal2.5', respectively.

Figure 8: In the provided segmentation outcomes, grey polygons depict the delineated crowns, points indicate the trees' locations, and coloring canopy class (red='Full sun', yellow='Partially shaded', blue='Mostly shaded'). The segmentation varies in panels (A) to (F), each corresponding to different quantile-derived window functions $\widehat{r}_\alpha$: 0.075, 0.125, 0.175, 0.225, 0.275, and 0.3, respectively. The coordinate axes in these plots adhere to the WGS84 coordinate system. Tree position data are incomplete.



similar to the one observed for the 'Dalp_stable_mini' in the simulation study. We suggest that the best result is achieved by $\alpha = 0.225$, which presents an F_1-score that balances between the two choices of stable windows. This choice also aligns with low jackknife standard deviation results displayed in Figure 6, in which lower values of the estimator were found for values of α close to 0.275. Further research should be done in this direction, with more informative data, to test these hypotheses. As a result, some tuning is necessary to choose the parameter α for applications in other regions where thinning of the ground measurements is present. We suggest looking for values less than 0.64, depending on the location's attributes and the initial dataset's thinning, as discussed in Appendix D. In cases where thinning is present, it would be very productive to have an estimation of the probability of thinning p, and tune α as discussed in Section 2.4.

Another noteworthy result is the robustness of the window size estimation, as depicted in Figure 6, under different training plots. The final remark concerns the decision to utilize the nearest neighbor distribution over other statistical measures from point process theory, like the empty space distribution or the *Ripley's K*-function. In essence, all these statistics exhibit strictly increasing functions w.r.t. distance, and as a result, they are all injective functions. Therefore, selecting a criterion for one statistic (e.g., $\alpha = 0.225$ for the NN distance distribution function) effectively implies an equivalent criterion when applied to another statistic, as shown in Section 2.4. This equivalency grants modelers complete flexibility in their choice of statistics.

As a final remark, we comment on the choice of the Hc model over the Poisson. Table E.4 shows that the estimated hard-core distance (r_0) is relatively small. Given this observation, a test for complete spatial randomness fails to reject the Poisson null hypothesis. Nevertheless, we argue that the Poisson model could serve as a computationally efficient approximation. We opted for the Hard-core model for two main reasons: (i) it enforces a biologically plausible constraint by disallowing arbitrarily close points, and (ii) it yielded slightly better segmentation performance in our experiments. A Poisson model assumption was used in Ene et al. (2012), where the logarithm of forest density was estimated via a linear model using two statistics of height measured by LiDAR as predictors: (i) the mean, and (ii) the coefficient of variation, both calculated over certain areas. A direct comparison of the two approaches could be explored in future work, as such a linear model cannot be reliably trained under incomplete measurements, as is the case in the real dataset we studied.

4.2. Limitations

The present study contains some limitations that need to be addressed. First, the incomplete dataset limits our understanding of the methodology's performance. To tackle this issue, (i) we conducted a simulation study under different types of forests, and (ii) we examine different values of α, as discussed in 2.4, and presented in Figures 6, 7. This gives us some insight into the accuracy, but does not address the potential over-segmentation issues in the real dataset application. Furthermore, the

argument discussed in Section 2.4, regarding the consideration of smaller values of α does not guarantee any convergence property to an optimal value. Extra caution is recommended when choosing the value of α in cases where the thinning probability p is unknown. Thinning applied to tree positions can also distort the estimation of the hardcore radius parameter, often leading to its overestimation. The magnitude of this bias depends on both the true values of (β, R) and the thinning probability. Estimation under high thinning probabilities should therefore be approached with caution.

Another limitation of the current approach is its supervised nature. Even though ground truth data are always necessary for validation purposes, getting rid of this limitation is a problem we look forward to addressing in the future.

A further constraint is the stationarity assumption of the tree position process. It is unfeasible to test the stationarity with only one sample of the point process (Illian et al., 2008; Chiu et al., 2013). As a result, we chose to proceed with this simplifying assumption, which in many cases is a reasonable one (Cox and Isham, 1980; Chiu et al., 2013). We have to comment that when the homogeneity assumption is violated, parameter estimation for the Hc model may be compromised—similar cases are discussed in Baddeley et al. (2015). This issue could be addressed by incorporating models that account for intensity variation w.r.t. exogenous variables, such as elevation, groundwater content, or solar radiation, or other factors w.r.t. the domain of application.

A remaining challenge lies in the segmentation scheme adopted in this study, which does not implement a multistage framework, i.e., sequentially segmenting dominant, sub-dominant, and suppressed trees. This hierarchical strategy presents a promising direction for future research, as similar approaches have been previously investigated Duncanson et al. (2014).

Finally, more segmentation algorithms exist in the bibliography, which were not included in our analysis. Examples of such algorithms include Convolutional Neural Networks (CNNs) (Chehreh et al., 2023), methods that include RGB data (Yancho et al., 2019), or voxel-based methods (Wang et al., 2016).

5. Conclusion

In the present study, we refined the LM method for tree segmentation by incorporating principles from point process theory, focusing mainly on optimizing the search window size. We aimed to validate the approach under simulated and real-world conditions and assess its performance across different canopy classes. The methodology successfully produces various adaptive window sizes, under incomplete plot measurements with clear and intuitive interpretations, leading to improved segmentation accuracy compared to existing methods. Despite these promising results, further research must address computational efficiency and scalability limitations. This study offers a significant methodological contribution, achieving state-of-the-

art results that open avenues for enhancing tree segmentation techniques in complex forestry environments.

6. Acknowledgements

The authors would like to thank the NEON organization for the availability of well-documented data products and express their gratitude to Marios Lardas for assistance in the creation of the graphical abstract.

7. Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT (v. 4o, o1) to improve the manuscript's readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Appendix A. Pseudolikelihood

The initial idea of the pseudolikelihood was introduced in Besag (1975), first applied to random fields. Before we present the pseudolikelihood function, the notion of *Papangelou conditional intensity* is needed.

Definition 3. *Let X be a point process on S with density f_θ, where θ a proper parameter. The function*

$$\lambda_\theta(\xi \mid \boldsymbol{x}) = \begin{cases} \dfrac{f_\theta(\boldsymbol{x} \cup \xi)}{f_\theta(\boldsymbol{x})}, & \xi \in S \setminus \boldsymbol{x}, \quad \text{if } f_\theta(\boldsymbol{x}) > 0, \\ 0, & \xi \in S \setminus \boldsymbol{x}, \quad \text{if } f_\theta(\boldsymbol{x}) = 0. \end{cases} \tag{A.1}$$

is called Papangelou conditional intensity *Kallenberg (1983)).*

The initial definition of the pseudolikelihood is achieved through a limit formulation that can be found in Besag (1975); Moller and Waagepetersen (2003). In Jensen and Moller (1991), the authors derived the following simplified formula

$$PL_B(\theta \mid x) = \exp\left(-\int_B \lambda_\theta(\xi \mid x)\, d\xi\right) \prod_{\xi \in x_B} \lambda_\theta(\xi \mid x \setminus \xi), \tag{A.2}$$

where $\lambda_\theta(\cdot \mid \mathbf{x})$ denotes the Papangelou conditional intensity associated with the density of the process, and B is the bounded observation set.

Appendix B. Dao-Genton correction

The motivation for the Dao-Genton test (Dao and Genton, 2014) is derived as follows. To ensure that the classical Bootstrap Goodness-of-Fit (GOF) hypothesis test is correctly sized, meaning its empirical level, $\hat{\alpha}$, matches the nominal level, α, the objective is to determine $\hat{\alpha}^*$ such that

$$\mathbb{P}(\hat{P} < \hat{\alpha}^* \mid H_0 \text{ is true}) = \alpha, \tag{B.1}$$

where $\hat{\alpha}^*$ is an estimate of α^*, known as the adjusted level, and $\hat{P}$ is the random variable associated with $\hat{p}$, the p-value from a single bootstrap hypothesis test. Typically, α^* is unknown, and $\alpha^* = \alpha$ only in the case of Complete Spatial Randomness, i.e., for the Poisson process. The decision rule of rejecting H_0 if $\hat{p} \leq \hat{\alpha}^*$ results in a correctly sized test because

$$\mathbb{P}\left(\text{Reject } H_0 \text{ at level } \hat{\alpha}^* \mid H_0 \text{ is true}\right) = \alpha \tag{B.2}$$

as indicated by equation (B.1). The adjusted GOF test consists of three steps.

The first step is to determine $\hat{p}$. Initially, B' samples from H_0 are used to estimate the theoretical values of the NN distribution function

$$\widehat{G}_{\text{theo}}(r\,;\theta_0) = \frac{1}{B'}\sum_{b=1}^{B'} \widehat{G}_b(r\,;\theta_0) \tag{B.3}$$

and hence obtain the Kolmogorov-Smirnov (K-S) distance of the empirical NN distribution function $\widehat{G}$ of the original sample from the theoretical one $\widehat{G}_{\text{theo}}$:

$$u = \|\widehat{G} - \widehat{G}_{\text{theo}}\|_\infty.$$

Then, another B samples are generated to calculate the bootstrap sample of the KS-distances of the empirical NN distribution function $\widehat{G}_b$ of each simulated sample from the previously estimated theoretical one:

$$u_b = \|\widehat{G}_b - \widehat{G}_{\text{theo}}\|_\infty, \quad b = 1, \ldots, B.$$

~~where $\widehat{G}$ and $\widehat{G}_b$ are the empirical NN distribution functions of the original sample and of sample b, respectively.~~ So $\hat{p}$ can be calculated as

$$\hat{p} = 1 - \frac{1 + \sum_{b=1}^{B} \mathbb{1}_{\{u_b \leq u\}}}{B+1}.$$

The second step is to find $\hat{\alpha}^*$. This is done by reconstructing the $\hat{P}$ distribution via pseudorandom sampling using a Monte Carlo technique. The following procedure

provides the pseudovalues, $\hat{p}_1, \ldots, \hat{p}_B$ for $\hat{P}$. They estimate the unknown p-values, $p_1, \ldots, p_B$, of the hypotheses

$$H_{i,0} : X_i \sim f_{\hat{\theta}_b}, \tag{B.4}$$

where $\hat{\theta}_b$ is the estimated parameters of the sample b. To compute $\hat{p}_1, \ldots, \hat{p}_B$, we define

$$\widehat{G}_{\text{theo}}(r\,; \hat{\theta}_b) = \frac{1}{B'} \sum_{b=1}^{B'} \widehat{G}_b(r\,; \hat{\theta}_b). \tag{B.5}$$

and

$$u_{b,0} = \|\widehat{G}_{\text{theo}}(\cdot\,; \hat{\theta}_b) - \widehat{G}_b\|_\infty \tag{B.6}$$

$$u_{i,j} = \|\widehat{G}_{\text{theo}}(\cdot\,; \hat{\theta}_b) - \widehat{G}_{i,j}\|_\infty, \quad i, j = 1, \ldots, B, \tag{B.7}$$

where $\widehat{G}_{i,j}$ is the empirical NN of the (i, j) sample. Finally, we estimate the p-value for sample b

$$\hat{p}_b = 1 - \frac{1 + \sum_{j=1}^{B} \mathbb{1}\{u_{b,j} \leq u_{b,0}\}}{B + 1}, \tag{B.8}$$

and find $\hat{\alpha}^*$ so that

$$\frac{1}{B} \sum_{b=1}^{B} \mathbb{1}_{\{\hat{p}_i < \hat{\alpha}^*\}} = \alpha.$$

The third step is to reject H_0 if $\hat{p} \leq \hat{\alpha}^*$, where $\hat{\alpha}^*$ is the α-quantile of $\hat{p}_1, \ldots, \hat{p}_B$.

Appendix C. Empirical Estimators

Let X be a stationary point process in $S \subseteq \mathbb{R}^d$, x_B be an observation from X in a window B, and

$$d(\xi, A) := \inf\{\|\xi - \xi'\| : \xi' \in A\} \tag{C.1}$$

be the distance of a point $\xi \in \mathbb{R}^d$ to a set A. The *uncorrected* estimator of the NN distribution function is

$$\widehat{G}(r) = \frac{1}{n(x)} \sum_i \mathbb{1}\{d_i \leq r\}, \quad r > 0, \tag{C.2}$$

where $d_i = d\,(\xi_i, x_B \setminus \{\xi_i\})$ the observed distance to the NN of the point $\xi_i \in x_B$. This estimator is biased (Moller and Waagepetersen, 2003; Illian et al., 2008; Badde-

ley et al., 2015), since observed points inside the observation window B could have an unobserved NN outside B. For this reason, corrected estimators have been proposed. Here, we present the *Kaplan-Meier* estimator of the NN distribution function (Baddeley and Gill, 1997), as is the one used in the text. For more information, the reader is directed to Moller and Waagepetersen (2003); Illian et al. (2008); Baddeley et al. (2015). In this case, we also denote by $b_i = d(\xi_i, \partial B)$, where ∂B denotes the boundary of the observation window B. Then, the *Kaplan-Meier* estimator is defined as

$$\widehat{G}_{KM}(r) = 1 - \prod_{\substack{d \in D \\ d \leq r}} \left(1 - \frac{n\big(\{i : d_i = d,\, d_i \leq b_i\}\big)}{n\big(\{i : d_i \geq d,\, d \leq b_i\}\big)} \right), \quad r > 0, \tag{C.3}$$

where $D = \{d_1, d_2, \ldots, d_n\}$.

Appendix D. Optimal value for PPP

Let us consider two point processes. First X a PPP with intensity λ and second X_{hc} the hardcore process generated from X by removing the points closer than the hardcore radius, similar to Al-Hourani et al. (2016). Then, it is straightforward

$$G_{X_{hc}} \leq G_X, \tag{D.1}$$

which validates that an optimal choice of α, noted as $\alpha^{\star}_{\text{PPP}}$ for the PPP, is an upper bound for the NN distance distribution function of the Hc process, resulting in an upper bound for the range of considerations. To calculate such an optimal $\alpha^{\star}_{\text{PPP}}$ consider the Ripley's K-function, equation (8), of the PPP:

$$K(r) = \pi r^2. \tag{D.2}$$

Asking for

$$\begin{aligned} K(r_1) = 1/\lambda \Rightarrow \\ \lambda \pi r_1^2 = 1, \end{aligned} \tag{D.3}$$

is equivalent to asking for one point per ball of radius r_1, see equation (8). Recall that the NN distance distribution function of the PPP is:

$$G_X(r) = 1 - \exp(-\lambda \pi r^2), \tag{D.4}$$

for which the value r_1 along with equation (D.3) yields

$$\alpha^{\star}_{\text{PPP}} = G_X(r_1) = 1 - \exp(-1) \approx 0.63. \tag{D.5}$$

This result indicates that the optimal $\alpha^{\star}_{\text{PPP}}$ for the PPP is bounded above by 0.64.

Appendix E. Estimated parameters for all heightclasses

References

Al-Hourani, A., Evans, R.J., Kandeepan, S., 2016. Nearest neighbor distance distribution in hard-core point processes. IEEE Communications Letters 20, 1872–1875. doi:10.1109/LCOMM.2016.2591017.

Alabama State Climate Summary, 2023. https://statesummaries.ncics.org/chapter/al/. Accessed: 17/12/24.

Baddeley, A., Gill, R.D., 1997. Kaplan-meier estimators of distance distributions for spatial point processes. The Annals of Statistics , 263–292doi:10.1214/aos/1034276629.

Baddeley, A., Rubak, E., Turner, R., 2015. Spatial point patterns: methodology and applications with R. CRC press. doi:10.1201/b19708.

Baddeley, A., Turner, R., 2014. Bias correction for parameter estimates of spatial point process models. Journal of Statistical Computation and Simulation 84, 1621–1643. doi:10.1080/00949655.2012.755976.

Beer, 1852. Bestimmung der absorption des rothen lichts in farbigen flüssigkeiten. Annalen der Physik 162, 78–88.

Besag, J., 1975. Statistical analysis of non-lattice data. Journal of the Royal Statistical Society Series D: The Statistician 24, 179–195. doi:10.2307/2987782.

Bowman, M.M., 2020. Linking Soil Organic Matter Composition to Soil Physical and Chemical Properties Across a Diverse Set of Soils: A Multi Method Approach. Ph.D. thesis. University of Colorado at Boulder.

Chehreh, B., Moutinho, A., Viegas, C., 2023. Latest trends on tree classification and segmentation using uav data—a review of agroforestry applications. Remote Sensing 15, 2263. doi:10.3390/rs15092263.

Chiu, S.N., Stoyan, D., Kendall, W.S., Mecke, J., 2013. Stochastic geometry and its applications. John Wiley & Sons. doi:10.1002/9781118658222.

Cox, D.R., Isham, V., 1980. Point processes. volume 12. CRC Press. doi:10.1201/9780203743034.

Daley, D.J., Vere-Jones, D., 2003. An introduction to the theory of point processes: volume I: elementary theory and methods. Springer. doi:10.1007/b97277.

Daley, D.J., Vere-Jones, D., 2008. An introduction to the theory of point processes: volume II: general theory and structure. Springer. doi:10.1007/978-0-387-49835-5.

Daley, N.M., Burnett, C.N., Wulder, C., Niemann, K.O., Goodenough, D.G., 1998. Comparison of fixed-size and variable-sized windows for the estimation of tree crown position, in: IGARSS'98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings.(Cat. No. 98CH36174), IEEE. pp. 1323–1326. doi:10.1109/IGARSS.1998.691394.

Dalponte, M., Coomes, D.A., 2016. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. Methods in ecology and evolution 7, 1236–1245. doi:10.1111/2041-210X.12575.

Dao, N.A., Genton, M.G., 2014. A monte carlo-adjusted goodness-of-fit test for parametric models describing spatial point patterns. Journal of Computational and Graphical Statistics 23, 497–517. doi:10.1080/10618600.2012.760459.

Dereudre, D., 2019. Introduction to the theory of gibbs point processes. Stochastic Geometry: Modern Research Frontiers , 181–229doi:10.1007/978-3-030-13547-8_5.

Duncanson, L., Cook, B., Hurtt, G., Dubayah, R., 2014. An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems. Remote Sensing of Environment 154, 378–386.

Ene, L., Næsset, E., Gobakken, T., 2012. Single tree detection in heterogeneous boreal forests using airborne laser scanning and area-based stem number estimates. International Journal of Remote Sensing 33, 5171–5193. doi:10.1080/01431161.2012.657363.

Eysn, L., Hollaus, M., Lindberg, E., Berger, F., Monnet, J.M., Dalponte, M., Kobal, M., Pellegrini, M., Lingua, E., Mongus, D., et al., 2015. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space. Forests 6, 1721–1747. doi:10.3390/f6051721.

Florakis, K., 2025. Simulated lidar realisations over simulated forest scenes. URL: https://https://zenodo.org/records/15924384, doi:https://doi.org/10.5281/zenodo.15924384. dataset.

Gao, T., Gao, Z., Sun, B., Qin, P., Li, Y., Yan, Z., 2022. An integrated method for estimating forest-canopy closure based on uav lidar data. Remote Sensing 14, 4317. doi:10.3390/rs14174317.

Georgii, H.O., 2006. Canonical Gibbs measures: some extensions of de Finetti's representation theorem for interacting particle systems. volume 760. Springer. doi:10.1007/BFb0068557.

Hatchner, C., 2017. Neon site-level plot summary: Talladega national forest (tall). NEON (National Ecological Observatory Network). USDA.

Hyyppa, J., Kelle, O., Lehikoinen, M., Inkinen, M., 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, 969–975. doi:10.1109/36.921414.

Illian, J., Penttinen, A., Stoyan, H., Stoyan, D., 2008. Statistical analysis and modelling of spatial point patterns. John Wiley & Sons. doi:10.1002/9780470725160.

Jaakkola, A., Hyyppä, J., Yu, X., Kukko, A., Kaartinen, H., Liang, X., Hyyppä, H., Wang, Y., 2017. Autonomous collection of forest field reference—the outlook and a first step with uav laser scanning. Remote sensing 9, 785. doi:doi.org/10.3390/rs9080785.

Jensen, J.L., Künsch, H.R., 1994. On asymptotic normality of pseudo likelihood estimates for pairwise interaction processes. Annals of the Institute of Statistical Mathematics 46, 475–486. doi:10.1007/BF00773511.

Jensen, J.L., Moller, J., 1991. Pseudolikelihood for exponential family models of spatial point processes. The Annals of Applied Probability 1, 445–461. doi:10.1214/aoap/1177005877.

Kaartinen, H., Hyyppä, J., Yu, X., Vastaranta, M., Hyyppä, H., Kukko, A., Holopainen, M., Heipke, C., Hirschmugl, M., Morsdorf, F., et al., 2012. An international comparison of individual tree detection and extraction using airborne laser scanning. Remote Sensing 4, 950–974. doi:10.3390/rs4040950.

Kallenberg, O., 1983. Random measures. De Gruyter. doi:10.1515/9783112471746-010.

Khosravipour, A., Skidmore, A.K., Isenburg, M., Wang, T., Hussin, Y.A., 2014. Generating pit-free canopy height models from airborne lidar. Photogrammetric Engineering & Remote Sensing 80, 863–872. doi:10.14358/PERS.80.9.863.

Krause, K., Goulden, T., 2022. Neon l0-to-l1 discrete return lidar algorithm theoretical basis document (atbd). NEON.DOC.001292vB. NEON (National Ecological Observatory Network).

Lai, M.J., Schumaker, L.L., 2007. Spline functions on triangulations. volume 110. Cambridge University Press. doi:10.1017/CBO9780511721588.

Lambert, J.H., 1760. Photometria sive de mensura et gradibus luminis, colorum et umbrae. sumptibus vidvae E. Klett, typis CP Detleffsen.

Li, W., Guo, Q., Jakubowski, M.K., Kelly, M., 2012. A new method for segmenting individual trees from the lidar point cloud. Photogrammetric Engineering & Remote Sensing 78, 75–84. doi:10.14358/PERS.78.1.75.

Ma, K., Chen, Z., Fu, L., Tian, W., Jiang, F., Yi, J., Du, Z., Sun, H., 2022. Performance and sensitivity of individual tree segmentation methods for uav-lidar in multiple forest types. Remote Sensing 14, 298. doi:10.3390/rs14020298.

Mase, S., 1995. Consistency of the maximum pseudo-likelihood estimator of continuous state space gibbsian processes. The Annals of Applied Probability 5, 603–612. doi:10.1214/aoap/1177004697.

Meier, C., 2022. Tos protocol and procedure: Vst – measurement of vegetation structure. NEON.DOC.000987vK. NEON (National Ecological Observatory Network).

Møller, J., Vihrs, N., 2022. Should we condition on the number of points when modelling spatial point patterns? International Statistical Review 90, 551–562. doi:10.1111/insr.12501.

Moller, J., Waagepetersen, R.P., 2003. Statistical inference and simulation for spatial point processes. CRC press. doi:10.1201/9780203496930.

Monnet, J.M., Mermin, E., Chanussot, J., Berger, F., 2010. Tree top detection using local maxima filtering: a parameter sensitivity analysis, in: 10th International Conference on LiDAR Applications for Assessing Forest Ecosystems (Silvilaser 2010), pp. 9–p.

Mueller, J., 2007. Arbaro: Tree generation software. https://arbaro.sourceforge.net/. Source code available at https://github.com/wdiestel/arbaro. Accessed: 2025-06-09.

National Forests in Alabama, 2024. https://www.fs.usda.gov/detail/alabama/about-forest/districts/?cid=fsbdev3_002556. Accessed: 17/12/24.

NEON, 2023a. Discrete return lidar point cloud (dp1.30003.001). URL: https://data.neonscience.org/data-products/DP1.30003.001/RELEASE-2023, doi:10.48443/XXBY-5A18.

NEON, 2023b. Vegetation structure (dp1.10098.001). URL: https://data.neonscience.org/data-products/DP1.10098.001/RELEASE-2023, doi:10.48443/73ZN-K414.

Picos, J., Bastos, G., Míguez, D., Alonso, L., Armesto, J., 2020. Individual tree detection in a eucalyptus plantation using unmanned aerial vehicle (uav)-lidar. Remote Sensing 12, 885. doi:10.3390/rs12050885.

Popescu, S.C., Wynne, R.H., 2004. Seeing the trees in the forest. Photogrammetric Engineering & Remote Sensing 70, 589–604. doi:10.14358/PERS.70.5.589.

Popescu, S.C., Wynne, R.H., Nelson, R.F., 2002. Estimating plot-level tree heights with lidar: local filtering with a canopy-height based variable window size. Computers and electronics in agriculture 37, 71–95. doi:10.1016/S0168-1699(02)00121-7.

PRISM Climate Data Explorer, 2023. https://prism.oregonstate.edu/explorer/. Accessed: 17/12/24.

Puliti, S., Breidenbach, J., Astrup, R., 2020. Estimation of forest growing stock volume with uav laser scanning data: can it be done without field data? Remote Sensing 12, 1245. doi:10.3390/rs12081245.

Quenouille, M.H., 1949. Approximate tests of correlation in time-series 3, in: Mathematical Proceedings of the Cambridge Philosophical Society, Cambridge University Press. pp. 483–484. doi:10.1017/S0305004100025123.

Reitberger, J., Schnorr, C., Krzystek, P., Stilla, U., 2009. 3d segmentation of single trees exploiting full waveform lidar data. ISPRS Journal of Photogrammetry and Remote Sensing 64, 561–574. doi:10.1016/j.isprsjprs.2009.04.002.

Ripley, B.D., 1977. Modelling spatial patterns. Journal of the Royal Statistical Society: Series B (Methodological) 39, 172–192. doi:10.1111/j.2517-6161.1977.tb01615.x.

Ripley, B.D., 1988. Statistical inference for spatial processes. Cambridge university press. doi:10.1017/CBO9780511624131.

Roussel, J.R., Auty, D., Coops, N.C., Tompalski, P., Goodbody, T.R., Meador, A.S., Bourdon, J.F., De Boissieu, F., Achim, A., 2020. lidr: An r package for analysis of airborne laser scanning (als) data. Remote Sensing of Environment 251, 112061. doi:10.1016/j.rse.2020.112061.

Silva, C.A., Hudak, A.T., Vierling, L.A., Loudermilk, E.L., O'Brien, J.J., Hiers, J.K., Jack, S.B., Gonzalez-Benecke, C., Lee, H., Falkowski, M.J., et al., 2016. Imputation of individual longleaf pine (pinus palustris mill.) tree attributes from field and lidar data. Canadian journal of remote sensing 42, 554–573.

Solberg, S., Naesset, E., Bollandsas, O.M., 2006. Single tree segmentation using airborne laser scanner data in a structurally heterogeneous spruce forest. Photogrammetric Engineering & Remote Sensing 72, 1369–1378. doi:10.14358/PERS.72.12.1369.

Tonolli, S., Dalponte, M., Neteler, M., Rodeghiero, M., Vescovo, L., Gianelle, D., 2011. Autonomous collection of forest field reference—the outlook and a first step with uav laser scanning. Remote Sensing 3, 2247–2262. doi:10.3390/rs9080785.

USGS Mineral Resources Data System (MRDS), 2023. https://mrdata.usgs.gov/mrds/. Accessed: 17/12/24.

Vauhkonen, J., Ene, L., Gupta, S., Heinzel, J., Holmgren, J., Pitkänen, J., Solberg, S., Wang, Y., Weinacker, H., Hauglin, K.M., et al., 2012. Comparative testing of single-tree detection algorithms under different types of forest. Forestry 85, 27–40. doi:10.1093/forestry/cpr051.

Wallace, L., Lucieer, A., Watson, C.S., 2014a. Evaluating tree detection and segmentation routines on very high resolution uav lidar data. IEEE Transactions on Geoscience and Remote Sensing 52, 7619–7628. doi:10.1109/TGRS.2014.2315649.

Wallace, L., Musk, R., Lucieer, A., 2014b. An assessment of the repeatability of automatic forest inventory metrics derived from uav-borne laser scanning data. IEEE Transactions on Geoscience and Remote Sensing 52, 7160–7169. doi:10.1109/TGRS.2014.2308208.

Wang, Y., Hyyppä, J., Liang, X., Kaartinen, H., Yu, X., Lindberg, E., Holmgren, J., Qin, Y., Mallet, C., Ferraz, A., et al., 2016. International benchmarking of the individual tree detection methods for modeling 3-d canopy structure for silviculture and forest ecology using airborne laser scanning. IEEE Transactions on Geoscience and Remote Sensing 54, 5011–5027. doi:10.1109/TGRS.2016.2543225.

Wang, Y., Lehtomäki, M., Liang, X., Pyörälä, J., Kukko, A., Jaakkola, A., Liu, J., Feng, Z., Chen, R., Hyyppä, J., 2019. Is field-measured tree height as reliable as believed–a comparison study of tree height estimates from field measurement, airborne laser scanning and terrestrial laser scanning in a boreal forest. ISPRS journal of photogrammetry and remote sensing 147, 132–145. doi:10.1016/j.isprsjprs.2018.11.008.

Wasserman, L., 2006. All of nonparametric statistics. Springer Science & Business Media. doi:10.1007/0-387-30623-4.

White, J.C., Tompalski, P., Vastaranta, M., Wulder, M.A., Saarinen, N., Stepper, C., Coops, N.C., 2017. A model development and application guide for generating an enhanced forest inventory using airborne laser scanning data and an area-based approach. Information Report FI-X-018. Natural Resources Canada, Canadian Forest Service, Canadian Wood Fibre Centre. Victoria, BC, Canada. doi:10.13140/RG.2.2.26770.96964.

White, J.C., Wulder, M.A., Varhola, A., Vastaranta, M., Coops, N.C., Cook, B.D., Pitt, D., Woods, M., 2013. A best practices guide for generating forest inventory attributes from airborne laser scanning data using an area-based approach. The Forestry Chronicle 89, 722–723. doi:10.5558/tfc2013-132.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., et al., 2019. Welcome to the tidyverse. Journal of open source software 4, 1686. doi:10.21105/joss.01686.

Winiwarter, L., Esmorís Pena, A.M., Weiser, H., Anders, K., Martínez Sánchez, J., Searle, M., Höfle, B., 2022. Virtual laser scanning with helios++: A novel take on ray tracing-based simulation of topographic full-waveform 3d laser scanning. Remote Sensing of Environment 269. URL: https://www.sciencedirect.com/science/article/pii/S0034425721004922, doi:https://doi.org/10.1016/j.rse.2021.112772.

Wood, S., Wood, M.S., 2015. Package 'mgcv'. R package version 1, 729.

Yancho, J.M.M., Coops, N.C., Tompalski, P., Goodbody, T.R., Plowright, A., 2019. Fine-scale spatial and spectral clustering of uav-acquired digital aerial photogrammetric (dap) point clouds for individual tree crown detection and segmentation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12, 4131–4148. doi:10.1109/JSTARS.2019.2942811.

Table E.4: Estimated parameters of the Hc process for all the height classes considered, as discussed in Section 2.4. First column represents the height class, while 2nd and third column represent the intensity (β_0) and the *hardcore distance* (r_0).

height class	β_0	r_0
≥ 2	0.043	0.584
≥ 2.5	0.042	0.613
≥ 3	0.042	0.613
≥ 3.5	0.041	0.613
≥ 4	0.041	0.613
≥ 4.5	0.041	0.613
≥ 5	0.041	0.613
≥ 5.5	0.041	0.613
≥ 6	0.041	0.613
≥ 6.5	0.04	0.613
≥ 7	0.039	0.613
≥ 7.5	0.039	0.613
≥ 8	0.039	0.613
≥ 8.5	0.039	0.613
≥ 9	0.039	0.613
≥ 9.5	0.038	0.613
≥ 10	0.037	0.613
≥ 10.5	0.036	0.621
≥ 11	0.035	0.62
≥ 11.5	0.034	0.62
≥ 12	0.033	0.62
≥ 12.5	0.032	0.62
≥ 13	0.031	0.62
≥ 13.5	0.029	0.62
≥ 14	0.028	0.619
≥ 14.5	0.026	0.619
≥ 15	0.025	0.619
≥ 15.5	0.023	0.619
≥ 16	0.021	0.618
≥ 16.5	0.02	0.617
≥ 17	0.019	0.617
≥ 17.5	0.019	0.617
≥ 18	0.018	0.616
≥ 18.5	0.016	0.616
≥ 19	0.016	1.107
≥ 19.5	0.016	1.106
≥ 20	0.016	1.22