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], Véronique Letort[c], Raphaël Canals[a], Gilles Faÿ[c], Samis Trevezas[b]

[a] *University of Orléans, Château de la Source, Orléans, 45100, , France*
[b] *National and Kapodistrian University of Athens, Panepistimioupolis 84, Zografou, 15784, , Greece*
[c] *CentraleSupélec - Université Paris-Saclay, 3 Rue Joliot Curie, Gif-sur-Yvette, 91190, , 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. Although most of the segmentation algorithms in the literature are designed to be unsupervised, some hyperparameters, which are likely to be site- or species-specific, need to be set by the users, thus increasing the risk of bias. In this direction, based on point process theory, we introduce formal implementation guidelines to refine the window size selection for the class of Local Maxima algorithms, a widely adopted class of methods for tree segmentation. This methodology can also be applied to incomplete plot measurements. We apply this methodology to an open dataset to ensure the reproducibility of the results. The method achieves an F_1-score in the $55 - 90\%$ range, depending on the positioning of the tree within the canopy relative to its neighboring trees, i.e., its dominance status.

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 **??**. These sophisticated technologies facilitate high-resolution, three-dimensional representations of forested landscapes, offering unprecedented detail and accuracy (**???**). The ability to accurately segment individual trees from LiDAR data is pivotal for enhancing our understanding of forest structure and dynamics and for implementing effective conservation and management practices **?**. However, this task presents unique challenges due to the intricate nature of forest canopies and the voluminous data generated by LiDAR systems (**???**). Furthermore, while current methodologies

produce globally satisfactory results for dominant trees, with precision greater than 85% in **?**, for most methodologies and plot types, studies such as **???** report the need for algorithms capable of satisfactorily segmenting even trees that exist in the second canopy layer. These include the canopy classes of intermediate and suppressed trees, partially and completely overshadowed, respectively. For instance, **?** 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. **?** tested algorithms developed in recent years, such as **??**, and 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 circ. 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 (**?**). 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 **?????** 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 fixed (stable) or variable, i.e., *adjusted* with respect to (w.r.t.) tree/forest attributes, such as height, species, or crown diameter (**?**).

- **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. **?** 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, **?** 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. **?** have adapted the method from **?** 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, **?** apply Point Cloud Detection and Delineation with a 2 m radius over a CHM smoothed again with a Gaussian filter. In more recent years, **?** 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 **?** 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 **?** 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 **??** in the early 2000s, who estimated the adjusted window size by using a linear regression model that modeled the maximum crown radius with respect to powers of height. This approach was notably implemented in the study by **?**, where the window size was determined as a function of height with coefficients calculated in previous research by **?**. Similarly, **?** 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 context of the application. 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 **????**. In their benchmark, **?** 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 **?**. Note that, however, their results are not differentiated per canopy class.

An application of adjusted window based on tree height was also implemented in **???**, and later implemented by **?**, 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 **?**, 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: Ene and colleagues introduced a technique rooted in the Poisson point process in **?**. 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 (**?**).

The following section provides a description of the dataset used in this study. Subsequently, a concise theoretical overview of the underlying theory applied in this research is presented. A hypothesis test is conducted 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. Dataset

Data used in this study were downloaded from the National Science Foundation's National Ecological Observatory Network (NEON) collection. Ground truth and LiDAR data come from **?** and **?**. A summary of those can be found in Table 1. Information on data collection procedures and protocols can be found on **?** and **?**, respectively.

The area of interest extends over 43 km^2 in the Talladega National Forest (TALL), in west-central Alabama (US) (32.95, -87.39) (Figure 1). The typical climate is subtropical, with an average annual temperature of 17.2°C and an average annual precipitation of 1380 mm (**??**). 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 (**??**). 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 (**??**). 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 2 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 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 (see Figure 1). 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 0.5×0.5 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 (**?**), for the range of heights $0, 3, \ldots, 24$ m. The sub-circle parameter of the algorithm was set to 0.4 m.

Table 1: 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 **???**, 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

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

Table 2: 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)

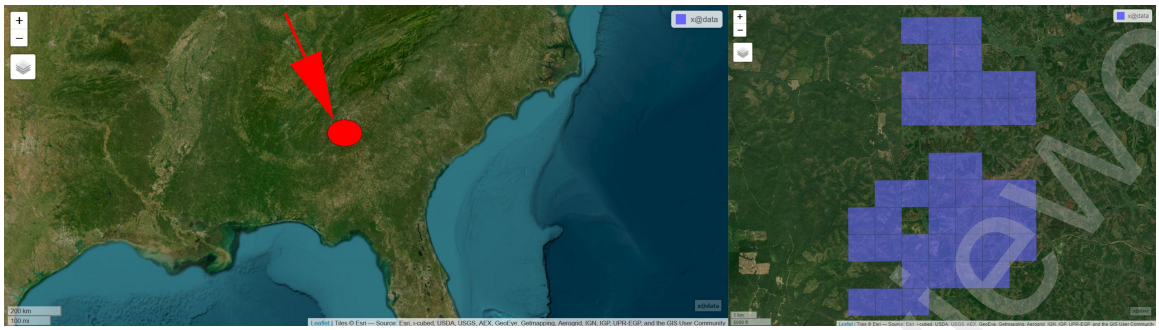


Figure 1: Location map. Each blue square denotes an area of $1\,km^2$, totaling 43 squares.

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.

2.2. 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). The observed points are in a bounded observation window in our application context.

To formalize the ideas above and justify our modeling approach, we introduce the necessary mathematical background and establish some notation in this section. For this purpose, we follow principally **?**. For a comprehensive exploration of point processes, the reader is also directed to **?????**.

2.2.1. 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 $\mathcal{B}(\mathbb{R}^2)$.

Definition 1. *Let $B, S \in \mathcal{B}(\mathbb{R}^2)$ be such that $B \subset S$, f be a probability density function on B, and n be a positive integer. A point process X consisting of n inde-*

pendent and identically distributed (i.i.d.) points with density f is called binomial point process *of n points in B with density f. We write $X \sim \mathsf{binomial}(B, n, f)$.*

In the simplest case of a bounded set B, with Lebesgue measure (volume) $\lambda(B)$, by selecting a constant density $1/\lambda(B)$, the binomial point process corresponds to a sample of uniformly distributed points on B.

Counting the number of points of a point process X within the measurable subsets of S is crucial to 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 2. *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 3. *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 $N(B) = n \in \mathbb{N}^*$, $X_B \sim \mathsf{binomial}(B, n, f)$ 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 a binomial point process of n points uniformly distributed on B.

2.2.2. Gibbs Processes

An intuitively more appealing model in the forest modeling context is based on the Hard-core (Hc) process, a particular case of Gibbs processes. The resulting models are crucial in understanding interactions within spatial point processes, particularly where individual points affect each other by either repulsion or attraction. This

interaction concept is significant in various scientific fields, such as spatial ecology, where organisms may compete for resources or territory (**?**).

Gibbs processes' density is defined w.r.t. the PPP (**??**). For example, the Hc process is a PPP under the restriction that points cannot be closer than a specified distance, a parameter of the process.

Let us now consider the case of a PPP with a constant intensity $\rho(\xi) = \rho > 0$. Following **?**, consider $B, S \in \mathcal{B}(\mathbb{R}^2)$ with $B \subseteq S$.

The joint density function corresponding to exactly n points in B, where the i-th point of the process is at (exactly) ξ_i, $i = 1, 2, \ldots, n$, is given by

$$\begin{aligned} f(n, \xi_1, \xi_2, \ldots, \xi_n) &= \frac{(\rho \cdot \lambda(B))^n}{n!} \exp\{-\rho \cdot \lambda(B)\} \cdot \prod_{i=1}^{n} \frac{1}{\lambda(B)} \\ &= \frac{\rho^n}{n!} \exp\{-\rho \cdot \lambda(B)\}. \end{aligned} \tag{3}$$

The expression (3) is valid for all $n \in \mathbb{N}$ and for all $(\xi_1, \ldots, \xi_n) \in B^n$, where by convention for $n = 0$ no points are present.

Now, notice that the terms in (3) are independent of $\{\xi_i\}_{i=1}^n$. In order to introduce more general processes, one can allow the existence of functions $g_n(\xi_1, \ldots, \xi_n)$ for each n, which are symmetric in their arguments. The joint density function corresponding to exactly n points in B, where the i-th point of the process is at ξ_i, $i = 1, 2, \ldots, n$, is given by

$$f(n, \xi_1, \xi_2, \ldots, \xi_n) = c\, \rho^n\, g_n(\xi_1, \ldots, \xi_n) \exp\{-\rho \cdot \lambda(B)\}/n!, \tag{4}$$

where c is a normalizing constant, determined by integrating the above equation w.r.t. $\xi_1, \ldots, \xi_n$ and summing over n. In the case of the homogeneous PPP $g_n(\xi_1, \ldots, \xi_n) = 1$. Note that

$$\mathbb{P}(N(B) = n) = \frac{c\, \rho^n\, e^{-\rho \cdot \lambda(B)}}{n!} \int_{B^n} g_n(\xi_1, \ldots, \xi_n)\, d\xi_1 \ldots d\xi_n, \tag{5}$$

and hence that, given $N(B) = n \geq 1$, the joint conditional density f_n is proportional to g_n, i.e.,

$$f_n(\xi_1, \ldots, \xi_n) := \frac{f(n, \xi_1, \xi_2, \ldots, \xi_n)}{\mathbb{P}(N(B) = n)} \propto g_n(\xi_1, \ldots, \xi_n). \tag{6}$$

Concerning the normalizing constant c, by (4), it is clear that it can be incorporated into the functions g_n and it plays no role in f_n. Additionally, since $g_n \geq 0$, the following representation is always valid

$$g_n(\xi_1, \ldots, \xi_n) = \exp\{-\phi_n(\xi_1, \ldots, \xi_n)\}, \tag{7}$$

as long as we formally allow the value $+\infty$ for ϕ_n to account for the null values of g_n.

Physicists refer to ϕ_n as a potential function of the process, and the overall process with joint density function given by (4) is referred to as *Gibbs process* with finite points. By (6) and the representation of g_n given by (7) we deduce that

$$f_n(\xi_1,\ldots,\xi_n) = \exp\big\{-\phi_n(\xi_1,\ldots,\xi_n)\big\}/\int_{B^n}\exp\big\{-\phi_n(\xi_1',\ldots,\xi_n')\big\}d\xi_1'\ldots d\xi_n'. \quad (8)$$

Until now, the point process has not been subject to any significant limitations. However, special families of the point processes are generated if we demand that all ϕ_n have similar structures induced by specific functions that model the multi-way interactions. One restriction is to assume that

$$\begin{aligned}\phi_n(\xi_1,\ldots,\xi_n) = n\psi_1 + \sum_{i_1>i_2}\psi_2(\xi_{i_1}-\xi_{i_2}) + \sum_{i_1>i_2>i_3}\psi_3(\xi_{i_1}-\xi_{i_2},\xi_{i_1}-\xi_{i_3})\\ +\ldots+\sum_{i_1>\cdots>i_s}\psi_s(\xi_{i_1}-\xi_{i_2},\ldots,\xi_{i_1}-\xi_{i_s}),\end{aligned} \quad (9)$$

where ψ_i are assumed to be symmetric in their arguments to ensure the same property for ϕ_n. In particular, $\psi_2(\xi) = \psi_2(-\xi)$ represents the potential contribution from a pair of points with distance vector ξ and allows the simplest interactions (pairwise). This formulation is neither arbitrary nor a convention (**?**). In physics, it is motivated by the goal of maximizing entropy (w.r.t. f_n) under a fixed expected total energy constraint

$$\int_B\cdots\int_B\phi_n(\xi_1,\ldots,\xi_n)f_n(\xi_1,\ldots,\xi_n)\,d\xi_1,\ldots,d\xi_n. \quad (10)$$

A simplification of the representation is possible. Consider the term $g_n(\xi_1,\ldots,\xi_n)\rho^n$ in (4). By (7) and the potential assumption (9) we yield the intensity term $\exp(-n\psi_1)\rho^n$. For each choice of ψ_1, we can modify appropriately ρ and get the same result. It is convenient to set $\psi_1 = 0$ and work only with ρ.

The simplest case, but still interesting enough, corresponds to the case in which only pairwise interactions are present. In this case, the potential is constructed by aggregating contributions from all possible pairs of points and can be expressed as follows

$$\phi_n(\xi_1,\ldots,\xi_n) = \sum_{i>j}\psi(\xi_i-\xi_j). \quad (11)$$

Further restrictions are needed to ensure that (5) defines a proper probability distribution over $\mathbb{N}$. Sufficient conditions can be found in **?**, but for a more comprehensive treatment, the reader is directed to **??**.

2.2.3. Hard-core Point Process (Hc)

The Hc process is a special case of a Gibbs process. The general form of the Gibbs process joint density is described by (8). The potential function ϕ_n corresponding to the Hc process is given by (11), thus allowing only pairwise interactions modeled by

$$\psi(\xi) = \begin{cases} \infty, & \text{if } \|\xi\| \leq R, \\ 0, & \text{if } \|\xi\| > R, \end{cases} \tag{12}$$

where R is a parameter of the process, the so-called *hardcore radius*. This parameter specifies a prohibitive distance within which points can not 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

$$f(\mathbf{x}_B) \propto \begin{cases} \beta^{n(\mathbf{x}_B)} & \text{, if } \|\xi - \xi'\| > R \text{ for all } \xi, \xi' \in \mathbf{x}_B,\ \xi \neq \xi', \\ 0 & \text{, if } \|\xi - \xi'\| \leq R \text{ for some } \xi, \xi' \in \mathbf{x}_B, \end{cases} \tag{13}$$

where $\theta = (\beta, R) \in \mathbb{R}^2_+$ are the parameters of the model, the intensity outside the hardcore radius, denoted as β instead of ρ for this model, and the hardcore radius, respectively. In this case, $\mathbf{x}$ represents a realization of the point pattern in a set-theoretic notation. Assuming a finite parameter β and 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 (**??**).

An intuitive yet inefficient method to simulate a Hc point process involves generating a PPP and eliminating points within the specified distance R (**???**). A more efficient way is to implement a Metropolis-Hastings algorithm (**?**).

2.2.4. 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

$$G(r) := \mathbb{P}\left(\inf_{x \in X \setminus \{u\}} \|x - u\| \leq r \,\middle|\, u \in X \right), \qquad r > 0. \tag{14}$$

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

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

discussion on the empirical estimation of the NN distribution function can be found in Appendix Appendix C. Henceforth, any reference to the empirical estimator of the NN distribution function will specifically denote the version incorporating the Kaplan-Meier correction.

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

$$K(r) := \frac{1}{\rho}\,\mathbb{E}\left[N\big(B(u,r)\setminus\{u\}\big)\;\middle|\; u \in X\right], \qquad r > 0, \tag{15}$$

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.

2.2.5. 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 (**??**). Nevertheless, parameter estimation can be performed by maximizing the pseudo-likelihood (PL) (**???**) (see Appendix Appendix A). Even though the maximum pseudo-likelihood estimator (MPLE) is biased (**?**), it is asymptotically unbiased, consistent, and asymptotically normal under appropriate regularity conditions (**???**).

An edge correction strategy must be implemented to eliminate *edge effect bias* (**?**). In **?**, 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 (**?**). 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 (**?**).

2.2.6. Train and test sets

Training and test sets 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).

2.2.7. 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 aim at 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 nearest neighbor 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 (**??**). For completeness, we present the entire algorithm in Algorithm 1, which we now describe.

Since a closed formula for the NN distribution function does not exist or a numerical approximation is needed (**?**), samples were used to estimate the theoretical values. These samples will be conditionally generated to match the number of points in the observed pattern. For the NN distribution function, this technique has proven to yield narrower envelopes, resulting in more robust outcomes (**?**).

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 the simulations, a *Metropolis-Hastings* algorithm was used. The theoretical values of the nearest neighbor distribution for the Hardcore process, denoted as $\widehat{G}_{\text{theo}}$, were 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 $1 - \frac{1}{B+1} \approx 0.995$, provided that $\widehat{G}$ lies entirely inside the confidence band, or equivalently if $\widehat{G}(r) \leq \widehat{G}_{\text{theo}} + D_{\max}$ and $\widehat{G}(r) \geq \widehat{G}_{\text{theo}} - D_{\max}$ for all r, where $D_{\max} = \max_b \|\widehat{G}_{\text{theo}} - \widehat{G}_b\|_\infty$, the maximum distance from the theoretical values.

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 **?**, require numerical approximations.

This Bootstrap-based hypothesis test is known to be conservative (**??**) and can introduce bias in the subsequent inference. For this reason, we added an extra step to the algorithm, suggested by **?**. We present a simple version of this methodology and postpone a more detailed discussion for the Appendix 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

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

in **??**.

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

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 Appendix B.

2.3. 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, 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 (15) 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 (15) by inverting the Ripley's K-function. We put

$$\widetilde{r}(\mu) = K_{\theta_0}^{-1}(\mu/\lambda_0), \tag{16}$$

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 **??** and in practice (16) is solved by numerical inversion.

The resulting solution depends clearly on the choice of μ. As indicated above, the choice $\mu = 1$ would be rather intuitive and probably efficient if tree positions in a given plot were completely known (complete data scenario). However, typically, plot measurements do not include all the trees, and this is also true for our 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 (**?**). 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

$$G_{\theta_0}^{-1}(\alpha^\star) = K_{\theta_0}^{-1}(1/\lambda_0) = \widetilde{r}(1). \tag{17}$$

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

$$G_{pX_{hc}} \leq G_{X_{hc}}, \tag{18}$$

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

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

$$r_{\alpha^\star} = G^{-1}_{X_{hc}}(\alpha^\star). \tag{19}$$

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

$$\widehat{r}_{\alpha^\star} = G^{-1}_{pX_{hc}}(\alpha^\star). \tag{20}$$

In both cases

$$G_{pX_{hc}}(\widehat{r}_{\alpha^\star}) = G_{X_{hc}}(r_{\alpha^\star}) = \alpha^\star, \tag{21}$$

which, combined with (18), results in

$$\widehat{r}_{\alpha^\star} \geq r_{\alpha^\star}. \tag{22}$$

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.

2.3.1. 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 2.2.5 to the training dataset for these classes yields estimates for each class, resulting in a piecewise constant window width function with respect to height over the intervals $\{(h_i, h_{i+1})\}_{i=1,\ldots,n}$. Finally, a 6 knots cubic spline (**?**) 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 (**??**), and up to a maximum height of $h_n = 20$ m beyond which there are not enough data left.

2.4. Estimating errors

Window size estimators.. To estimate the error of various estimators derived from quantiles (as detailed in Equation (16)), the jackknife methodology (**?**) 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

$$b_{\text{jack}}(\widehat{r}_a) = (m-1)\left(\frac{1}{m}\sum_{i=1}^{m}\widehat{r}_a^{-i} - \widehat{r}_a\right), \tag{23}$$

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

$$s_{\text{jack}}(\widehat{r}_a) = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(\widetilde{r}_a(i) - \frac{1}{m}\sum_{j=1}^{m}\widetilde{r}_a(j)\right)^2}, \tag{24}$$

where $\widetilde{r}_a(i) = m\,\widehat{r}_a - (m-1)\,\widehat{r}_a^{-i}$ the so called pseudo-values (**?**). 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):

$$\frac{1}{N_{\text{poly}}}\sum_{i=1}^{N_{\text{poly}}}\mathbb{1}_{\{\text{exactly one tree in the polygon } i\}}, \tag{25}$$

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

$$\frac{1}{N_{\text{trees}}} \sum_{i=1}^{N_{\text{trees}}} \mathbb{1}_{\{\text{tree } i \text{ is uniquely contained within a polygon}\}}, \tag{26}$$

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

$$F_1 = 2 \cdot \frac{\text{sensitivity} \cdot \text{precision}}{\text{sensitivity} + \text{precision}}. \tag{27}$$

Only polygons with at least one authentic tree position were considered in the described analyses. Indeed, due to incomplete tree measurements within each plot (**?**), 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.

2.5. Benchmark approaches

Ground truth data was split into training and test datasets to test these accuracies as described in Section 2.2.6. 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 2.3, was included in a Dalponte2016 implementation (**?**). For comparison, we used a Watershed algorithm, similar to **?**, which is one of the most used algorithms in our (LiDAR) context (**?**), the Li2012 algorithm (**?**), two approaches similar to **?**, and three instances 3 cases of the Dalponte2016 algorithm (**?**) with fixed circular window sizes of $1, 1.5$, and 2.5 m radius.

2.5.1. 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 $0.75, 1, 1.5, 2, 2.5, 3$.

2.5.2. Li2012 algorithm

The method developed by **?** employs a 'top-down' approach for segmenting data, initiating segmentation from the 'global maxima' (GM) of an unsegmented 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 **?** 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 $d_1 = 1.5\,\mathrm{m}$, $d_2 = 2\,\mathrm{m}$, and $Z_u = 15\,\mathrm{m}$ were used, suggested by the original publication, as default values. The Li2012 algorithm is mainly used as a benchmark of a no-tuned algorithm.

2.5.3. Popescu approaches

We further included in the comparison two different cases of the approach used in **??**. 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*).

2.6. Implementation

The coding was done in R language (version 4.2.3 (2023-03-15 ucrt)). Most of the data processing and plotting was performed utilizing the *tidyverse* package (**?**). Algorithms for processing LiDAR data were implemented using the *lidR* package (**?**). Splines were implemented using the *mgcv* package (**?**). Spatial models were handled using the *spatstat* package (**?**).

3. Results

Figure 2 displays the hypothesis test results through confidence bands visualization, as presented in Section 2.2.7. 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.3. In Figure 3, estimations of radii for circular windows using various quantile estimators as outlined in Section 2.3 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 (23), (24)) are computed on the training dataset and presented in Figure 4. 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, $b_{\mathrm{jack}}(\widehat{r}_{0.275}) = 0.06$ (4 % rel. error) and $s_{\mathrm{jack}}(\widehat{r}_{0.275}) = 0.11$ (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 value of the

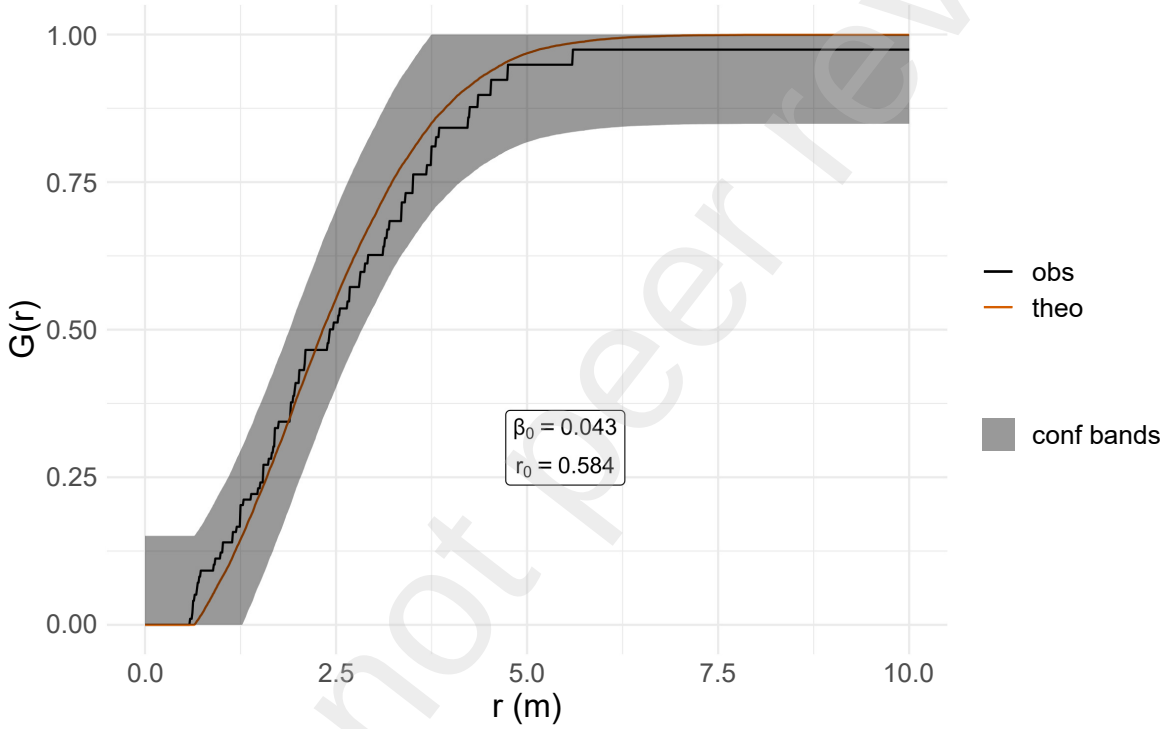


Figure 2: Dao-Genton envelopes (**?**) for the Hypothesis test as outlined in Section 2.2.7. The red line is the theoretical value of the nearest neighbor distribution of the Hardcore process under the null hypothesis, the black line is the observed one, and the grey area indicates the envelopes under H_0. The estimated parameters, under H_0, can be found inside the annotation box.

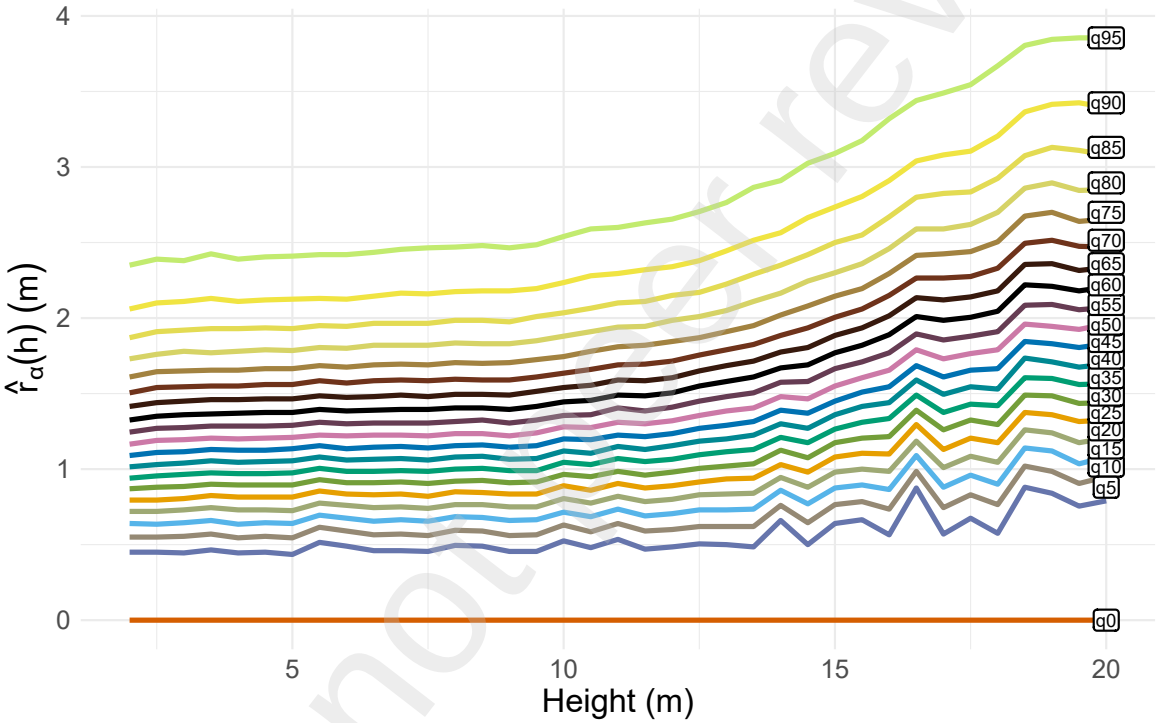


Figure 3: Circular window radius ($\widehat{r}_\alpha$) estimates for different quantile estimators, as a function of height (h), according to the methodology detailed in Section 2.3. Each line represents the window size radius in meters (m). Quantile probabilities start from 0 (bottom line) and increase by 0.05, as the annotations describe.

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 3, 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 5 presents the F_1-score results, as described in Section 2.4. The

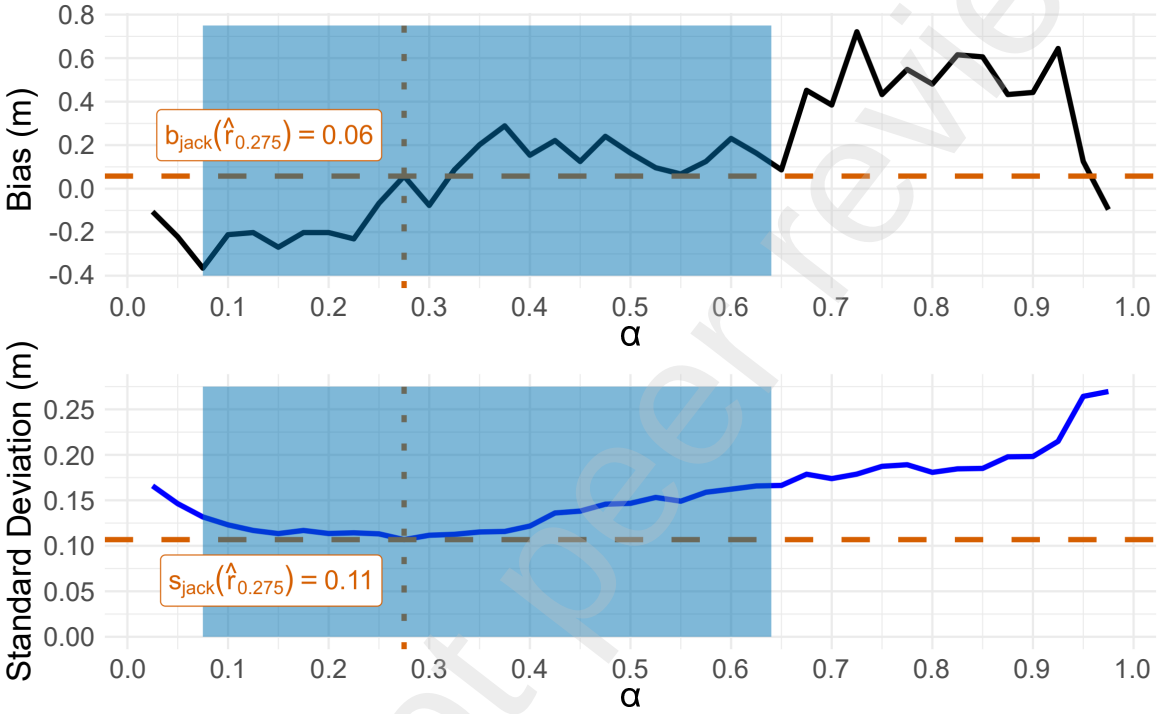


Figure 4: Different quantile estimators of window function $\widehat{r}_\alpha$ (x-axis) compared under the estimated jackknife bias (top) and standard deviation (bottom) (Section 2.4). Values of α range between $[0.025, 0.975]$ with a 0.025 step. The shaded area displays the α values in $[0.075, 0.64]$. The horizontal dashed lines indicate the achieved minimum w.r.t. the absolute jackknife bias and jackknife standard deviation, respectively, achieved for $\alpha = 0.275$.

curved line represents the different estimator w.r.t. α choice. Comparatively, dark blue illustrates the Watershed algorithm's performance, and light blue corresponds to the Li2012 algorithm. Yellow and dark green lines follow methodologies akin to those in **??** using pre-estimated parameters and with re-estimation, respectively.

Segmentation results can be observed in Figure 6. The selection of the specific plot ('TALL_007', as coded in the dataset **?**) was made to allow for 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.

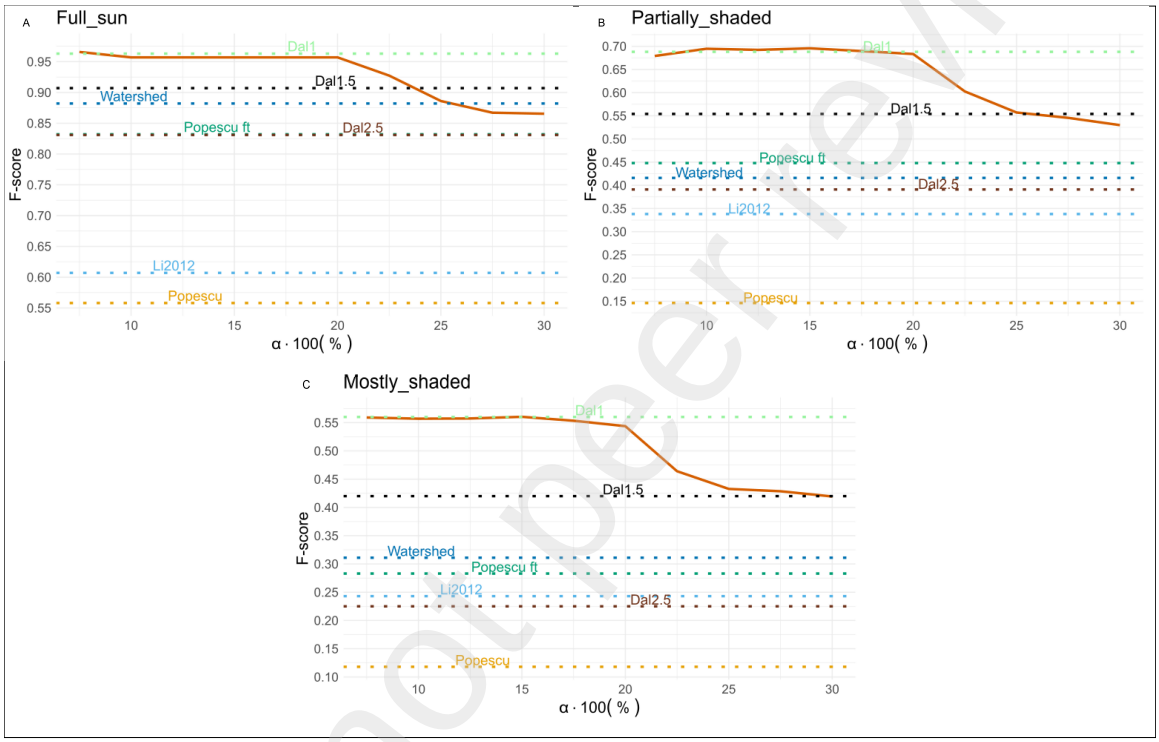


Figure 5: Results of Segmentation accuracy: (A) Only 'Full sun' (B) with the inclusion of 'Partially shaded' (Section 2.4) (C) with the further inclusion of 'Mostly shaded'. Different quantile estimators of the window function $\widehat{r}_\alpha$, as presented in Section 2.3 are compared under F_1-score criterion (27). 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 **??**, while dark green represents a similar methodology with a re-estimation of the parameters.

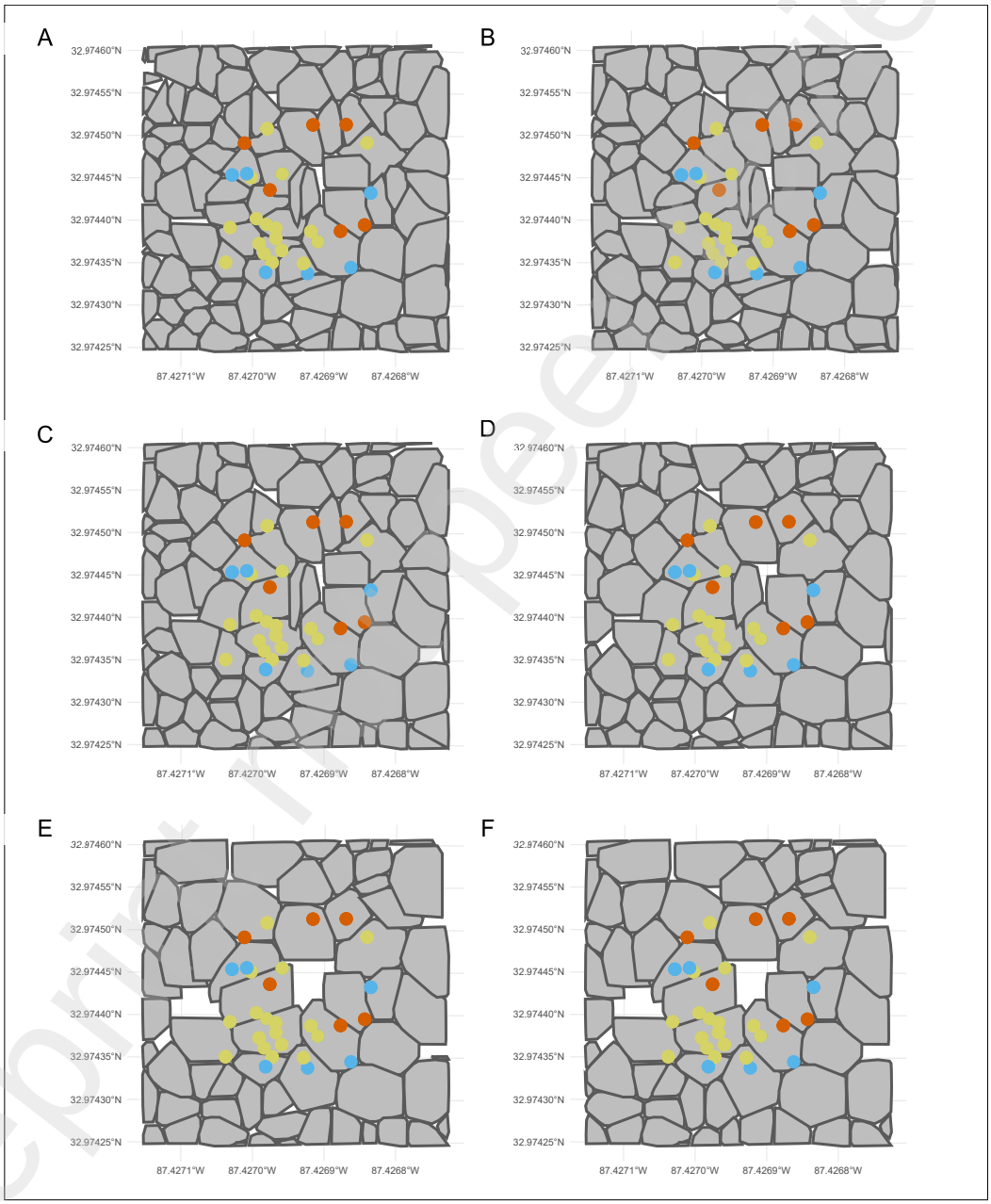
Clear misclassifications can be observed in the middle of the images, where some 'Partially shaded' trees are found in the same polygon in all the segmentation cases. The two northwest points are misclassified in the last two cases (E-F) but are correctly classified in the other segmentations. Another 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 current 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 and test it in different canopy class cases. The present findings also support the results reported by **???** regarding the lower accuracies for suppressed trees, as can be observed in Figures 5 and 6. Applying the hardcore process in this context yielded promising outcomes, indicating a good fit for the model, as depicted in Figure 2. The final 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, following the discussed approach, as can be observed in Figure 5. We attribute the latter results 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 w.r.t. different α parameters, as demonstrated in Figures 5 and 3. We think that for values of $\alpha \leq 0.2$, where the results are equivalent to the stable circular window of radius 1 m, over-segmentation is present. 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 4, 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 Appendix D. In the case thinning is present, it would be very productive to have an estimation of the probability of thinning p.

Figure 6: 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.



Another noteworthy result is the robustness of the window size estimation, as depicted in Figure 4, 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 2.3. This equivalency grants modelers complete flexibility in their choice of statistics.

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, we examine different values of α, as discussed in 2.3. This gives us some insights into the accuracy but does not address the potential oversegmentation issues. Furthermore, the argument discussed in Section 2.3, regarding the consideration of smaller values of α does not prove any convergence property to an optimal value. Extra caution is recommended when choosing the value of α in the cases where the thinning probability p is unknown.

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 (**??**). As a result, we chose to proceed with this simplifying assumption, which in many cases is a reasonable one (**??**).

Finally, more segmentation algorithms exist in the bibliography, which were not included in our analysis. Examples of such algorithms include Convolutional Neural Networks (CNNs) (**?**), methods that include RGB data (**?**), or voxel-based methods (**?**).

5. Conclusion

In the present study, we refined the LM method for tree segmentation by incorporating principles from point process theory, mainly focusing on optimizing the search window size. We aimed to validate the approach under 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.

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

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

Appendix A. Pseudolikelihood

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

Definition 4. *Let X be a point process on S with density f_θ, where θ... The function*

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

is called Papangelou conditional intensity **?***).*

The initial definition of the pseudolikelihood is achieved through a limit formulation that can be found in **??**. In **?**, the authors derived the following simplified formula

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

where $\lambda_\theta(\cdot \,|\, \mathbf{x})$ 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 (**?**) 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

$$\mathbb{P}(\hat{P} < \hat{\alpha}^*) = \alpha, \tag{B.1}$$

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

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

Then, another B samples are generated to calculate

$$u = \|\widehat{G}_{\text{theo}} - \widehat{G}\|_\infty, \tag{B.4}$$

$$u_b = \|\widehat{G}_{\text{theo}} - \widehat{G}_b\|_\infty, \quad b = 1, \ldots, B, \tag{B.5}$$

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

$$H_{i,0} : X_i \sim f_{\hat{\theta}_b}, \tag{B.6}$$

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

$$\widehat{G}_{\text{theo}}(r\,;\hat{\theta}_b) = \frac{1}{B'}\sum_{b=1}^{B'} \widehat{G}_b(r\,;\hat{\theta}_b). \tag{B.7}$$

and

$$u_{b,0} = \|\widehat{G}_{\text{theo}}(\cdot\,;\hat{\theta}_b) - \widehat{G}_b\|_\infty \tag{B.8}$$

$$u_{i,j} = \|\widehat{G}_{\text{theo}}(\cdot\,;\hat{\theta}_b) - \widehat{G}_{i,j}\|_\infty, \quad i,j = 1,\ldots,B, \tag{B.9}$$

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

$$\hat{p}_b = 1 - \frac{1 + \sum_{j=1}^{B} \mathbb{1}\{u_{b,j} \leq u_{b,0}\}}{B+1}, \tag{B.10}$$

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

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

where $d_i = d\left(\xi_i, \mathbf{x}_B \setminus \{\xi_i\}\right)$ the observed distance to the NN of the point $\xi_i \in \mathbf{x}_B$. This estimator is biased (**???**), 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 (**?**), as is the one used in the text. For more information, the reader is directed to **???**. 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 **?**. 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 (15), 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 (15). 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 approximately 0.63.

Appendix E. Estimated parameters for all heightclasses

Table E.3: Estimated parameters of the Hc process for all the height classes considered, as discussed in Section 2.3. 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