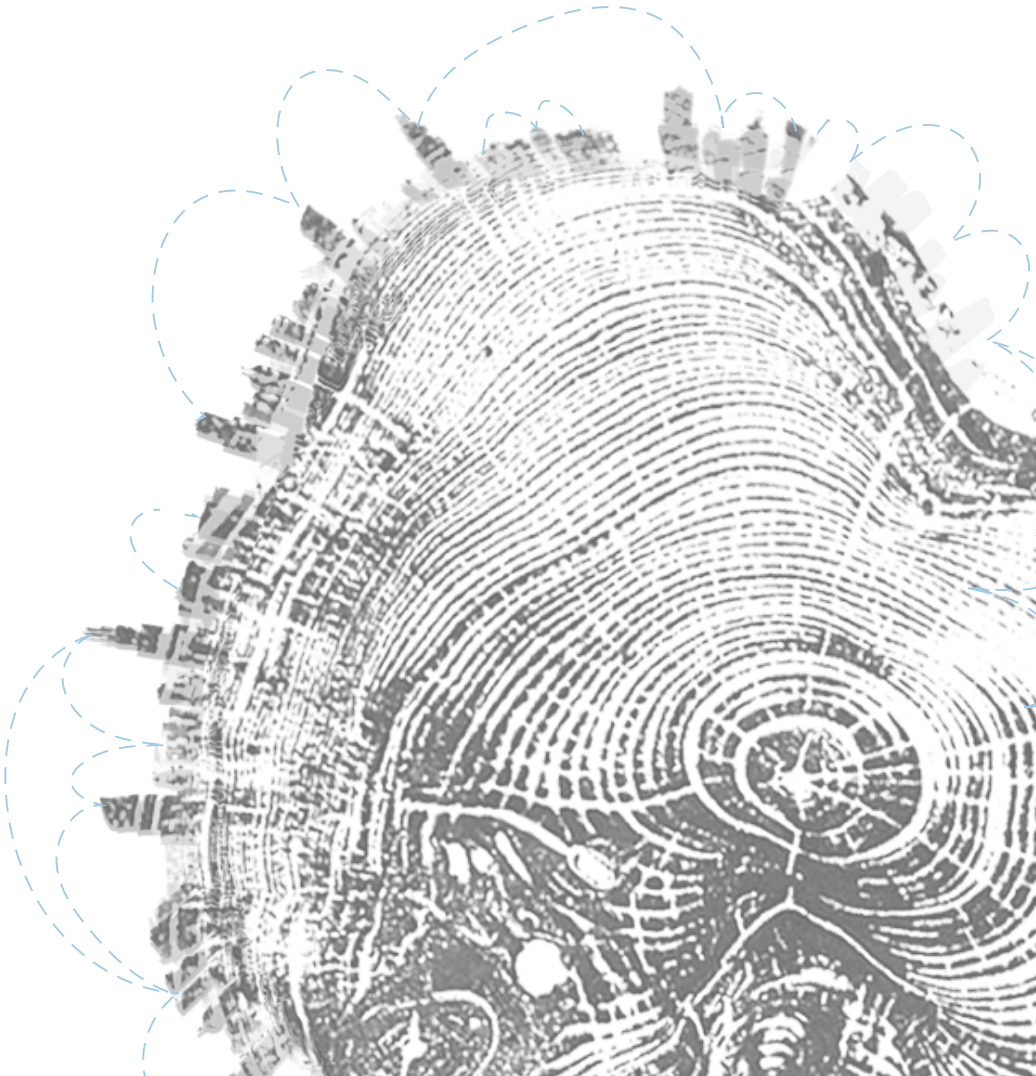
MSc Thesis in Geomatics

From Point Clouds to Porous Crowns: A Scalable Approach for CFD-Ready Urban Tree Reconstruction

Noah Petri Alting

2025



TU Delft

FROM POINT CLOUDS TO POROUS CROWNS:

A SCALABLE APPROACH FOR CFD-READY URBAN TREE RECONSTRUCTION

A thesis submitted to the Delft University of Technology in partial fulfilment
of the requirements for the degree of

Master of Science in Geomatics for the Built Environment

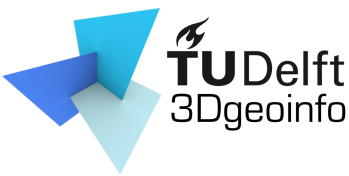
by

Noah Petri Alting

October 2025

Noah Petri Alting: *From Point Clouds to Porous Crowns: A Scalable Approach for CFD-Ready Urban Tree Reconstruction* (2025)

The work in this thesis was carried out in the:



3D geoinformation group
Delft University of Technology

Supervisors: Dr. Hugo Ledoux
Dr. Clara Garía Sánchez
Co-reader: Dr. Roderik Lindenbergh

ABSTRACT

Urban climate simulations increasingly rely on digital twins of cities, yet vegetation remains largely absent or oversimplified despite its strong influence on wind flow and heat exchange. Existing lidar-based tree studies mainly target forests or small plots, and no scalable method currently exists to reconstruct detailed urban trees for computational fluid dynamics (CFD) analysis.

This thesis presents a scalable, automated pipeline for reconstructing CFD-ready urban tree models from open-access airborne lidar. The workflow operates directly on unstructured point clouds and comprises three main components. The first introduces the *High-Order Multi-Echo Density (HOMED) vegetation filter*, a new approach for distinguishing vegetation from non-vegetation in airborne lidar point clouds. Combined with *TreeSeparation*, a cuboid-based tree instance segmentation algorithm, it produces clean per-tree clusters for subsequent analysis. The second explores taxonomy-based classification to assess structural separability between species. The third abstracts each segmented point cloud into CFD-suitable crown and trunk geometries and derives per-tree volumetric porosity values, ensuring watertightness, manifoldness, and controlled mesh complexity. Designed to be dataset-agnostic, the pipeline generalises to any comparable airborne laser scanning data.

The workflow was applied to several major Dutch cities—Amsterdam, Rotterdam, Utrecht, and Delft—processing hundreds of thousands of trees from raw point clouds to 3D models within practical runtimes ($\approx$ 13 hours for Amsterdam on 16 CPU cores). Segmentation and reconstruction proved robust and consistent, yielding α-wrapped crowns, cylindrical trunk approximations, and physically meaningful porosity estimates. Taxonomic classification, however, was unreliable due to label noise, class imbalance, and limited structural separability in lidar-only data; supplementary optical features from RGB and infrared imagery were evaluated but proved unreliable for consistent integration.

The results demonstrate that the pipeline enables scalable reconstruction of CFD-ready tree models that preserve canopy structure and aerodynamic properties, allowing their explicit representation in urban digital twins and supporting more realistic urban climate simulations.

ACKNOWLEDGEMENTS

This thesis marks the conclusion of my Master's in Geomatics at Delft University of Technology, and with it, for now, the end of my twenty-two years in education. Of all those years, this final one may have been the best. I truly loved working on this thesis and am proud to present this last piece in my academic puzzle.

I have never dedicated so much effort to anything before. Along the way, a group of wonderful people guided me, supported me, and at times carried me through. They deserve every bit of credit.

First, I want to sincerely thank my supervisors, Hugo and Clara. When I started looking for a thesis topic, I decided to try my luck with my two favourite teachers, hoping that one of them might have something related to nature — perhaps something about trees. Coincidentally, or perhaps fatefully, their research areas turned out to complement each other perfectly, allowing them to accommodate a topic that bridges two seemingly opposite worlds: the digital and the natural. Our easy-going discussions, filled with curiosity and good humour, made this process feel more like a creative journey than an academic obligation. Many students describe writing a thesis as a dreadful experience; I never felt that way, thanks to you.

Of course, not every day was bright and effortless. There were moments when I got stuck on designing code or understanding what came out. A big shoutout to Michalis (the Dionysian), Hidemichi (the Samurai) and Felix (the Oracle), my trusted allies in countless academic duels, for always being ready to help me unravel the tougher knots. Those coffee breaks in and around the library, conceptualising algorithms while looking at the trees surrounding us, made all the difference.

To my former housemates of JvB67: thank you for being my family away from home. I lived there through most of my student life, surrounded by a mix of personalities that somehow fit perfectly together. The older housemates were like the big brothers and sisters I never had, showing the way, offering advice, and setting an example. The housemates who were my age shared the same academic timeline, faced the same challenges, and now feel like my own season of Friends. And the ones who came after kept me young and, at times, took care of me when I forgot to take care of myself — even feeding me when I thought fasting would save time. A special mention goes to Storm, who, with his remarkably active spirit, made sure I did not forget to exercise or forget my love for the sea, and to Dorus, who has always been ready to talk about anything or nothing, day and night. To everyone I shared that wonderfully terrible house with: thank you for being such an essential part of my life.

Lastly, and probably most importantly, my parents. To my dad, thank you for always feeding my curiosity and teaching me to see the beauty in our world. It has been a while since you have "helped me with my homework", but know that I could not have done this without you. To my mom, words fall short of what you give me. Our talks ground me, guide me, and remind me of who I am. Emotionally and spiritually, we are inseparable. Together, you have shown me what it means to stay positive, enthusiastic, and independent — and that it is completely fine to sometimes not be. Your love, trust, and boundless support have made all of this possible.

Now, with this sentimental part addressed — let's get to the point.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3DBAG	3D Basisregistratie Adressen en Gebouwen
AOI	Area of Interest
AHN	Actueel Hoogtebestand Nederland
ALS	Airborne Laser Scanning
CFD	Computational Fluid Dynamics
CHM	Canopy Height Model
CNN	Convolutional Neural Network
CV	Cross-Validation
d50	Characteristic Tree Gap Size
DBH	Diameter at Breast Height
DNN	Deep Neural Network
DTM	Digital Terrain Model
F1	Harmonic Mean of Precision and Recall
GIS	Geographic Information System
HOMED	High-Order Multi-Echo Density
KD-tree	K-dimensional Tree
k-NN	k-Nearest Neighbours
LAD	Leaf Area Density
LAI	Leaf Area Index
LAS	Laser File Exchange Format
LoD	Level of Detail
NDVI	Normalised Difference Vegetation Index
NIR	Near Infrared
RANS	Reynolds-Averaged Navier–Stokes
RF	Random Forest
RGB	Red, Green, Blue
SOR	Statistical Outlier Removal
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
UHI	Urban Heat Island

1. INTRODUCTION

1.1. Urban Vegetation in Digital Twins and CFD Context

Urbanisation continues to intensify, with more people living in dense built environments. Since 2023, over 93% of the population in the Netherlands resides in urban areas (The World Bank, 2023). This trend increases the need for accurate digital tools to support sustainable and climate-resilient urban planning.

Three-dimensional city models are now widely used for this purpose, forming the backbone of digital twins that support decision-making on energy, noise, flooding, and urban climate (Biljecki et al., 2015). In the Netherlands, the 3DBAG dataset shows how far building modelling has advanced—offering automated, accurate, and detailed models of the national building stock at multiple levels of detail (Peters et al., 2022).

Trees, however, are largely absent from such models. Where they appear at all, they are often simplified as coarse porosity blocks or generic cylinders (García-Sánchez et al., 2021). Yet their influence on the urban microclimate is multifaceted: *aerodynamically*, trees modify wind flows and turbulence patterns, while *thermodynamically*, they regulate surface radiation, provide shading, and mitigate the urban heat island (UHI) effect. Recent work by Fu et al. (2024) demonstrated that the level of geometric detail in tree models can substantially affect simulated wind fields, confirming that oversimplified vegetation representations bias CFD-based urban climate predictions. Integrating vegetation into 3D city models is therefore essential for accurate simulations. While this thesis focuses on aerodynamic representation, the resulting high-resolution tree models are also directly applicable to radiation balance, shading, and urban heat island studies that rely on geometrically detailed vegetation inputs.

1.2. Challenges in lidar-based Urban Tree Modelling

Unlike buildings, which have rigid, planar surfaces, trees consist of porous, irregular structures with high shape variability, making them far more complex to represent geometrically and aerodynamically. Most existing lidar-based vegetation studies focus on relatively homogeneous settings such as forests or orchards. The review by Chehreh et al. (2023) highlights that most tree segmentation and classification research has been conducted on UAV-based datasets in agroforestry contexts, where vegetation cover is dense and relatively uniform. Similar forest-oriented approaches include Popov et al. (2024); Fu et al. (2020); Hell et al. (2022). While these methods achieve strong results in controlled settings, they are not directly transferable to complex urban environments.

Some studies target classification or reconstruction on single isolated trees (Pfeiffer et al., 2018; Rodriguez et al., 2024) or small experimental plots (Nowak et al., 2022; Keerthinathan et al., 2025), where occlusion and heterogeneity are limited. Others explore CFD for trees in small urban test areas (Somanath et al., 2024), but very few focus on large-scale reconstruction in heterogeneous urban settings. Efforts to classify tree taxa from lidar are also almost exclusively restricted to forest settings (Hell et al., 2022; Amiri et al., 2019; Slavík et al., 2023), and no existing work integrates species classification with large-scale geometric modelling.

Meanwhile, digital twin frameworks have made major strides in reconstructing buildings, yet they rarely incorporate trees at a comparable level of detail (Somanath et al., 2024). Where trees are included, they are often represented as bulk porosity fields or simplified primitives, overlooking their internal structure. However, CFD simulations require both porous canopies (to represent drag) and solid trunks (to represent near-ground flow obstruction) (Buccolieri et al., 2018).

Currently, no established method exists to reconstruct detailed 3D trees—including crown structure, trunk geometry, and per-tree porosity estimates—from airborne lidar at city scale. This lack of scalable tree modelling limits the realism of urban climate simulations.

1.3. Proposed Workflow for CFD-Ready Tree Model Reconstruction

This thesis develops a scalable, automated pipeline to reconstruct urban tree models from open-access AHN airborne lidar. While the approach is demonstrated on AHN data, it is designed to operate on any comparable airborne laser scanning dataset that provides multi-return information.

The pipeline operates directly on unstructured point clouds and consists of three main stages:

(i) filtering and segmenting individual tree instances, (ii) exploring taxonomy-based classification from tree structure in 3D lidar data, and (iii) reconstructing each tree into a simplified CFD-compatible model consisting of a crown, trunk, and porosity estimate.

Although taxonomic classification was initially explored to enable species-specific aerodynamic parameters, the limited separability of structural features in AHN data led the study to focus instead on per-tree geometric and porosity reconstruction. This yields physically grounded, species-agnostic inputs suitable for aerodynamic analysis in CFD simulations.

1.4. City-scale Implementation

The pipeline is designed to run at city scale by processing AHN tiles in parallel. Delft was used as the primary development site to tune the segmentation, porosity estimation, and geometry reconstruction steps, while Amsterdam, Rotterdam, and Utrecht provided additional labelled samples for classification experiments. The workflow scales efficiently across multiple CPU cores, enabling the reconstruction of large urban areas within practical runtimes.

1.5. Thesis Outline

This thesis is structured as follows. Chapter 2 reviews existing work and positions this study within the literature, followed by the research questions in Chapter 3. Chapter 4 outlines the strategy and overall workflow to answer the research questions. The three main research chapters then cover tree segmentation (Chapter 5), taxonomy-based classification (Chapter 6), and geometry reconstruction (Chapter 7), each ending with their own discussion and answer to the corresponding research question. Finally, Chapter 8 provides an overarching discussion and conclusions, synthesising the findings from all three main chapters.

2. RELATED WORK

This chapter reviews existing research relevant to the reconstruction of urban trees from airborne lidar. It is organised along the main components of the proposed pipeline: (i) how vegetation is represented in urban CFD workflows, (ii) how vegetation can be filtered from raw airborne lidar, (iii) how species have been inferred from point clouds, and (iv) how 3D tree geometries can be reconstructed. For each topic, the focus is on methods that could inform or contrast with the design choices of this thesis.

2.1. Vegetation in Urban CFD Modelling

Urban vegetation is a key driver of wind and heat behaviour, yet its representation in CFD models remains inconsistent and often oversimplified. This subsection reviews how vegetation is represented in computational fluid dynamics (CFD) studies and how aerodynamic concepts such as porosity and drag inform the geometric and semantic requirements of tree models.

2.1.1. Vegetation in Urban Digital Twins

Digital twins increasingly support urban climate simulations, yet vegetation is still commonly missing or replaced by generic objects. Understanding how trees are currently handled in such models highlights the baseline from which this thesis departs.

Vegetation is increasingly acknowledged as a relevant component in urban digital twin development, particularly for applications involving airflow and microclimate modelling. Somanath et al. (2024) present a procedural pipeline for generating 3D city models from geospatial data, aiming to bridge GIS formats with simulation-ready geometry.

Although their work focuses primarily on built structures, it underscores a broader limitation: current urban modelling workflows rarely incorporate vegetation in an end-to-end manner. Their case study relies on a coniferous forest and does not address urban tree types or species-specific structural variation. Nonetheless, it reinforces the aerodynamic relevance of vegetation, citing earlier work by García-Sánchez et al. (2021) showing that semantic differentiation—including vegetation—is essential for reliable CFD outcomes.

These studies collectively demonstrate that vegetation strongly influences wind and heat behaviour but is still largely omitted or simplified in current digital twin implementations. This highlights a clear gap in existing urban modelling pipelines regarding the explicit and detailed representation of vegetation.

2.1.2. Level of Detail and Vegetation Semantics in CFD

Representing trees for CFD involves both geometry and semantics. In this subsection, *semantics* refers to the surface classes that allow the solver to assign boundary conditions or source terms. In urban CFD these classes typically include ground, building (walls and roofs), water, and vegetation. Within vegetation, a further distinction is required between crowns and trunks because they are treated differently by the solver (crowns as porous media, trunks as solid obstacles). The geometric and aerodynamic requirements that complement these semantics, such as watertightness and per-tree porosity, are defined in Section 2.1.5.

The impact of LoD and semantics on urban CFD simulations is demonstrated by García-Sánchez et al. (2021), who simulated airflow across part of the TU Delft campus using city models at two

geometric levels (LoD 1.3 and 2.2), both with and without semantic surfaces for vegetation and water. Their results confirm that both geometric detail and surface classification (into ground, building, and vegetation surfaces) affect local wind patterns at pedestrian level, especially near tall buildings. Although their focus was on wind comfort, they note that similar effects are likely relevant for pollutant and pathogen dispersion, indicating a broader role for vegetation modelling.

These findings build on earlier work by Ricci et al. (2017), who showed that geometric simplification can distort flow behaviour in dense urban environments, and by Buccolieri et al. (2018), who emphasised the need for detailed Leaf Area Density (LAD) inputs in CFD to represent vegetation realistically. More recently, Fu et al. (2024) provided a quantitative assessment of how geometric simplification in tree representations affects urban microscale airflow simulations. By comparing multiple levels of detail (LoD 1–3) of identical trees, they demonstrated that even moderate reductions in geometric fidelity can alter predicted wind velocities and turbulence intensities near vegetation canopies. Taken together, these studies confirm that vegetation structure and geometric abstraction strongly influence aerodynamic outcomes in CFD and highlight the need for controlled, CFD-aware simplifications in reconstruction workflows.

The importance of distinguishing between different parts of a tree—such as modelling trunks as solid obstacles and crowns as porous media—has also been shown in CFD-based pollutant-dispersion studies. Hong et al. (2017), for example, explicitly modelled tree trunks as impermeable objects, demonstrating that this structural separation improves near-ground wind-flow predictions.

Taken together, these studies underscore that vegetation should be represented with both semantic and geometric detail to achieve reliable CFD outcomes. However, existing workflows lack a scalable way to translate lidar-based tree data into such detailed CFD-ready representations.

2.1.3. Porosity in CFD

Porosity is the key parameter controlling aerodynamic drag from vegetation in CFD. This part reviews how porosity is used in existing CFD studies and why realistic canopy porosity estimates are essential for accurate simulations.

In Computational Fluid Dynamics (CFD), vegetation is typically represented as a porous medium that modifies airflow through drag forces and turbulence generation. The detailed geometry of leaves and branches is rarely resolved explicitly due to computational cost. Instead, vegetation is parameterised through porosity or permeability values that determine how much air can pass through the canopy volume.

Two main notions of porosity are found in the literature. (i) *Optical or aerodynamic porosity* refers to the fraction of open space within a canopy, as seen visually or measured in wind tunnel experiments. (ii) *Volumetric porosity*, in contrast, is derived from point clouds or from Leaf Area Density (LAD) estimates, and is used in porous-medium CFD models. Both concepts allow vegetation resistance terms in the momentum equations to be calibrated, which directly influence canopy drag. The choice between them depends on the application: aerodynamic porosity is often used in windbreak design[1], while volumetric porosity is increasingly applied in urban CFD studies.

Expected porosity values vary widely with vegetation type and measurement method. Reviews of shelterbelts and windbreaks (Bitog et al. (2011); Brandle (2009)) typically report effective aerodynamic porosities between 0.4 and 0.7, with intermediate values giving the best wind reduction

[1] In agricultural contexts, windbreak design refers to the strategic planting of tree belts with specific porosities to reduce wind speed and protect fields. These studies provide a useful reference for expected aerodynamic porosity values.

while avoiding excessive turbulence. Optical porosity is strongly seasonal, ranging from about 0.1 at full leaf cover to 0.5 or more in winter conditions. In contrast, CFD studies that parameterise vegetation as porous zones often assume higher volumetric porosities ($\sim$0.8–0.95) to represent the bulk of the crown while still allowing significant air exchange through the foliage (Jeanjean et al. (2016); Lee et al. (2023)).

Taken together, these studies show that porosity is a critical parameter linking tree structure to aerodynamic behaviour in CFD. It forms the conceptual bridge between point-cloud-based estimates of canopy density and the resistance terms used in porous-medium CFD models. The next subsection reviews how such porosity values can be derived directly from lidar point clouds.

2.1.4. Tree Crown Porosities from Point Clouds

Several studies have attempted to estimate canopy porosity from lidar point clouds, usually through voxelisation. Reviewing these methods clarifies their sensitivity to voxel size and orientation, and highlights important limitations that need to be addressed for CFD applications.

Estimating tree porosity from point clouds is most commonly approached using voxel grids, which yield estimates of *volumetric porosity*. For example, Kamoske et al. (2019) present a method to extract 3D Leaf Area Density (LAD) grids and LAI–height curves through voxelisation combined with Beer–Lambert inversion. Their approach provides a strong framework for per-tree voxelisation from ALS data and for capturing height-stratified canopy structure. However, it does not directly yield the scalar porosity values required for CFD modelling of canopy geometries.

Nowak et al. (2022) apply voxelisation and voxel occupancy counts to map gaps along tree belts using ALS. Their work demonstrates how porosity can be derived at the level of tree groups, focusing on between-tree porosity. While a useful example of voxel-based porosity estimation, it does not address intra-crown volumetric porosity at the level of individual trees.

A different perspective is provided by Pfeiffer et al. (2018), who voxelised tree point clouds obtained from Terrestrial Laser Scanning (TLS). Their study highlights that the orientation of the voxel grid strongly influences canopy porosity estimates. Using a fixed 20 cm voxel size, they reported volumetric porosity values of 0.1570 (azimuthal), 0.0703 (frontal), and 0.2004 (lateral). From these values, I calculated an average porosity of 0.14 with a standard deviation of 0.05 ($\approx$38%). Notably, their use of the tree's convex hull as the bounding volume likely underestimates voxel occupancy by introducing voids outside the actual canopy.

Taken together, these studies demonstrate that voxelisation provides a viable method for deriving volumetric porosity scalars from tree point clouds. At the same time, they emphasise important limitations: porosity estimates are sensitive to point cloud density and occlusion effects (particularly in ALS data), as well as to voxel size and grid orientation. These constraints limit the robustness and transferability of current methods and must be addressed when developing CFD-ready canopy porosity models at the per-tree level.

2.1.5. CFD-Ready Tree Models and Requirements

Computational Fluid Dynamics (CFD) simulations solve airflow and turbulence numerically within a three-dimensional domain that represents the urban environment. Common approaches such as Reynolds-Averaged Navier–Stokes (RANS) or Large Eddy Simulation (LES) models discretise this domain into a volumetric mesh, where the accuracy and stability of the simulation depend on the quality of the input geometry. To prevent numerical leakage and ensure correct boundary enforcement, all geometries in the domain must be watertight and well-defined, as emphasised

by recent studies on mesh-driven CFD workflows and city-model preparation (Pađen et al., 2022; Jaiswal et al., 2024).

These requirements imply that CFD-ready models must consist of manifold, non-intersecting solids with clear surface boundaries. Overlapping or intersecting geometries can introduce non-manifold regions, which lead to meshing errors or solver divergence. In addition, overly detailed geometries can result in excessive mesh density and high computational cost, making them unsuitable for city-scale studies. Therefore, simplified but topologically consistent models with reasonable polygon counts are preferred for CFD workflows, with consistency in coordinate reference systems ensuring correct alignment between terrain, buildings, and vegetation layers. Pađen et al. (2022) further highlight the lack of vegetation representation in CFD-ready city models and identify the automatic reconstruction of trees from point clouds as a next step towards porous-zone modelling, referencing the approach of (de Groot, 2020).

CFD solvers assign physical properties and boundary conditions based on surface classes such as ground, building, vegetation, and water. These semantics determine how drag, roughness, or flow resistance is applied during the simulation. Within the vegetation class, a further distinction is required between trunks and crowns: trunks are treated as solid obstacles that block airflow, while crowns are represented as porous volumes that attenuate it (Buccolieri et al., 2018). Methods for deriving such solid and porous geometries are discussed in Sections 2.4.2 and 2.4.3. Additional aerodynamic parameters, such as drag coefficients, permeability, or leaf area density (LAD), are typically defined during CFD pre-processing rather than in the geometric model itself.

Based on these conventions, a CFD-ready tree model can be defined as a simplified, watertight geometry that distinguishes between solid trunks and porous crowns and can be assigned aerodynamic parameters required for CFD simulation. This thesis adopts these conventions to ensure that the reconstructed tree models are directly compatible with porous-flow CFD workflows and can serve as input for urban airflow or pollutant-dispersion studies.

2.2. Tree Delineation from Aerial Data

Before individual trees can be modelled, they must first be delineated—i.e., segmented as individual instances—from the surrounding urban environment. This subsection reviews different approaches to tree delineation from aerial data, including RGB imagery, airborne lidar, and combined lidar–hyperspectral datasets, and evaluates their suitability for urban contexts.

Accurate delineation is a prerequisite for representing vegetation as individual instances in CFD-ready city models. Treating vegetation as a single bulk zone tends to overestimate porosity by merging gaps between adjacent crowns and prevents the definition of discrete crown volumes or trunk positions. Segmentation into per-tree clusters ensures that each canopy's aerodynamic influence, crown geometry, and reconstructed trunk can be represented independently, establishing a semantically consistent foundation for higher-detail urban models and subsequent CFD parameterisation.

2.2.1. Delineation from RGB Aerial Imagery (Deep Learning)

Deep learning approaches can delineate trees from aerial imagery by training on labelled data to distinguish vegetation from non-vegetation areas.

Weinstein et al. (2020) present DeepForest, a deep learning framework for detecting individual trees from RGB aerial imagery using a pre-trained RetinaNet model. It outputs 2D bounding boxes of tree crowns and offers both pre-trained weights for immediate use and fine-tuning options

through annotation and custom training. DeepForest has shown strong performance for detecting dominant crowns in forested landscapes, where dense canopy cover and high-contrast textures support reliable detection.

However, its reliance on 2D imagery limits its ability to represent vertical structure and to separate overlapping crowns in complex urban scenes. Because it produces only planar bounding boxes, it provides less geometric detail than methods operating directly on 3D data. These constraints reduce its applicability for workflows that require structural descriptors or volumetric representations of canopy form.

Airborne lidar, by contrast, directly captures three-dimensional geometry and is available nationwide in uniform quality through the AHN dataset. Its structural measurements allow crown shapes and canopy densities to be derived without relying on external imagery, which in practice often varies in resolution and temporal consistency. These factors can complicate the use of colour drapes for segmentation. For these reasons, this study focuses exclusively on lidar-based segmentation.

2.2.2. Delineation from Airborne Lidar

Segmenting individual trees directly from airborne lidar point clouds remains challenging due to overlapping crowns, irregular canopy structures, and the sparse or occluded sampling typical of urban environments. Because lidar captures height and volume information explicitly, a wide range of three-dimensional algorithms have been developed to isolate individual tree crowns from these point clouds. The following paragraphs summarise the main families of such methods and assess their suitability for large-scale urban tree segmentation from ALS measurements.

Early airborne-lidar segmentation relied on two-dimensional *canopy height models* (CHMs), where local maxima were interpreted as tree tops and watershed algorithms delineated the surrounding crowns (e.g. Popescu et al., 2003; Chen et al., 2006; Jakubowski et al., 2013; Zhang et al., 2015). While effective for homogeneous forest stands, these raster-based methods discard vertical structure and tend to merge or truncate crowns in heterogeneous urban settings where canopy height changes abruptly near buildings. They also depend on grid resolution and smoothing parameters that are difficult to tune consistently across varying point densities. Moreover, because they operate on a 2D raster surface rather than directly on point clouds, such methods inevitably lose vertical information and produce unstable boundaries in complex urban scenes where trees neighbour buildings. For this reason, raster-based watershed approaches were not adopted here, as the aim was to maintain full 3D structure and consistent behaviour across heterogeneous AHN tiles.

To capture canopy structure more explicitly, later studies employed three-dimensional *voxel- and region-growing approaches* (e.g. Reitberger et al., 2009; Wu et al., 2013). These methods subdivide the point cloud into equally sized volumetric cells and cluster adjacent occupied voxels into candidate crowns. Although they represent height variation more realistically than CHM-based techniques, voxel approaches are sensitive to grid resolution and can incur high memory costs for large datasets. Overlapping crowns often remain merged unless voxel sizes are made so small that computation becomes prohibitive. More adaptive clustering schemes, such as the mean-shift algorithm of Hu et al. (2017), introduce dynamic kernel sizes to better capture local crown density variations, yet remain computationally demanding for large-area airborne datasets.

To overcome these issues, Wang et al. (2018) proposed a scalable tree-segmentation algorithm that replaces cubic voxels with vertically stretched *cuboids*. By refining the resolution horizontally while maintaining a coarser spacing vertically, the method achieves a finer separation of adjacent crowns without a proportional increase in computational load. A connectivity coefficient assigns

cuboids to trees, weighting face connections more strongly than edge or vertex connections, which improves delineation accuracy in dense canopies. The algorithm supports both top–down and bottom–up label propagation through the cuboid structure, making it adaptable across different lidar modalities. In their evaluation on a dataset of densely packed trees with manual ground truth, the method achieved a Cohen's kappa of 94 %, outperforming voxel-based baselines (89 %) and demonstrating robustness for overlapping crowns while remaining computationally efficient.

More recent work explores *learning-based segmentation* using neural-network architectures such as PointNet++ (Chen et al., 2021b). These models can reach very high accuracies when trained on dense terrestrial or UAV-based scans, yet they require extensive labelled data and significant computational resources that are not currently available for nationwide airborne datasets such as AHN. Consequently, their adoption for city-scale tree modelling remains limited.

Among these alternatives, the cuboid-based method provides the most pragmatic balance between accuracy, scalability, and reproducibility. Its anisotropic cell structure adapts naturally to varying crown sizes and pruned shapes, while its deterministic and unsupervised formulation avoids the need for training data. Because it is governed by only a few interpretable parameters, it can be tuned once and applied consistently across large urban areas. This robustness and computational efficiency make it particularly suitable for automated, lidar-only pipelines aimed at generating CFD-ready tree representations, where stable per-crown geometry is more critical than exact botanical boundaries.

2.2.3. Delineation from Lidar–Hyperspectral Fusion (Graph-Cut Method)

Fusing lidar with hyperspectral imagery can improve tree delineation by combining structural and spectral information, though it requires data types not available in the AHN dataset.

Lee et al. (2017) present a graph-cut-based method that delineates individual trees using airborne lidar and hyperspectral imagery. The algorithm constructs an energy function balancing spectral similarity and spatial coherence, which is then optimised through a min-cut/max-flow solution. This approach is particularly effective for isolating tall trees in forested areas, where strong height and spectral contrasts support clear canopy boundaries.

However, the authors note that its performance degrades for trees smaller than 20 meters, limiting its suitability for datasets with diverse urban tree sizes. While the method offers valuable insight into energy function optimisation for tree segmentation, its reliance on hyperspectral imagery and its reduced accuracy on smaller trees constrain its applicability to typical urban contexts.

2.3. Species Recognition from Airborne Lidar

Recognising tree species or broader taxonomic groups from airborne lidar has been an active topic in both forest and urban remote sensing. Species information inferred from three-dimensional structure supports diverse applications, including ecological monitoring, biomass estimation, and urban planning. The following section first outlines the machine-learning approaches most commonly applied, then summarises the geometric and spectral features used as model inputs, discusses how training labels have been obtained and curated in previous studies, and finally describes the botanical naming standards that enable consistent labelling across datasets.

2.3.1. Machine-Learning Approaches for Species Classification

Various machine-learning approaches have been explored to classify individual trees or broader tree types from airborne or UAV-based lidar point clouds. Most studies focus on homogeneous

forest environments, where training data are easier to obtain and crown structure is more uniform. This subsection reviews representative algorithms and highlights why Random Forest has become a common baseline for sparse airborne lidar.

Deep learning has been applied to point cloud data to improve accuracy and robustness. Chen et al. (2021a) proposed the Point Cloud Tree Species Classification Network (PCTSCN) to distinguish between white birch (broadleaf deciduous) and larch (coniferous). While limited in scope, this work demonstrates the feasibility of species-level classification from 3D structure.

Hell et al. (2022) applied deep neural networks (DNNs) to delineate four tree types—coniferous, deciduous, dead trees with crowns, and snags—achieving overall accuracies of 87.0% (PointCNN) and 73.2% (3DmFV-Net) using a point cloud density of 80 pts/m^2. A total of 2,721 trees were delineated using a normalised-cut segmentation method and manually labelled for training and evaluation. Incorporating features derived from multispectral orthophotos improved accuracies by up to 16.3%.

Recent reviews underscore that supervised learning models remain dominant in tree species classification tasks involving 3D aerial data. Chehreh et al. (2023) surveyed 144 works using UAV-derived data, noting that Random Forest is among the most widely used classifiers applied with point cloud or lidar-derived features. These studies often combine structural, height, density, and spectral features, and emphasise that performance depends heavily on feature engineering, sensor resolution, and preprocessing steps.

In the context of sparse airborne lidar, Random Forest offers advantages in interpretability and lower training data requirements compared to deep neural networks, making it a suitable candidate when data are limited or point density is moderate.

2.3.2. Structural and Spectral Features Used for Model Training

Building on these classification approaches, accurate species recognition depends strongly on the quality and relevance of features extracted from individual tree point clouds. These features capture geometric, structural, and foliage-related properties that can serve as discriminative inputs for machine learning models.

Commonly used geometric descriptors include tree height, crown width, crown depth, and convex hull volume, while structural features capture the vertical distribution of points or crown density profiles. Reviews such as Chehreh et al. (2023) highlight that these geometric and structural metrics are among the most frequently used predictors across studies, often combined with spectral information when available. Chi et al. (2025) provide a well-documented set of extractable structural features from tree point clouds, which are summarised in Appendix C.

Beyond static shape descriptors, more dynamic signatures can be derived from the distribution of leaf area along the vertical axis. Kamoske et al. (2019), also referenced in Parker (2020), propose generating LAI–height curves by summing LAD values per height bin. This produces a vertical "signature" of foliage density that may help distinguish species based on typical crown architectures and leaf distribution patterns.

Other foliage-related metrics, such as Leaf Area Index (LAI) and Leaf Area Density (LAD), have also been proposed as structural descriptors (Parker, 2020; Hermann, 2024; Buccolieri et al., 2018). Kamoske et al. (2019) demonstrate a voxel-based method to derive LAD from lidar point clouds and aggregate it into LAI profiles. While these metrics offer indirect cues on species and canopy density, many studies omit them when spectral data are unavailable, instead relying on geometric descriptors or derived volumetric measures such as canopy porosity.

2.3.3. Sources and Acquisition of Training Data Labels

Obtaining sufficient ground-truth data is a major limitation in tree species classification from lidar, as few datasets contain individual trees with verified species labels. Studies have therefore adopted a variety of strategies that balance spatial coverage against label reliability.

Most existing work still depends on visually delineated crowns and manually assigned species labels. In a comprehensive review, Chehreh et al. (2023) report that such reference data are commonly created for model evaluation rather than large-scale training, producing small but highly reliable datasets that are difficult to generalise across sites.

At the small scale, several studies rely on field-verified or visually interpreted samples. Chen et al. (2021a) collected all trees through on-site field identification, achieving very accurate labels but limiting the dataset to two species—white birch and larch—across a few plots. Hell et al. (2022) manually assigned class labels by inspecting each segmented tree's lidar point cloud together with aerial photographs. Their four classes—coniferous, deciduous, dead trees, and snags—illustrate how manual annotation often yields few, and sometimes subjective, categories but produces highly reliable labels for those categories, particularly in controlled forest settings.

To address the limited scale of such datasets, several studies link remotely sensed data with existing tree inventories. Chi et al. (2025) combined hyperspectral and lidar measurements with verified municipal and field inventories to train urban tree classifiers across multiple contexts, while de Groot (2020) demonstrated a fully lidar-based approach that spatially matched municipal tree records to segmented crowns. Inventory-based labelling provides access to far larger training sets and more taxonomically detailed classes than manual field surveys, but it introduces uncertainty in the spatial linkage between the recorded tree position and the corresponding segmented crown. Despite these challenges, municipal inventories remain valuable because their recorded locations and species names are generally stable over time, enabling consistent spatial linking with lidar data.

Together, these studies reveal a clear trade-off between label precision and dataset scale. Field surveys and manual annotation yield accurate but small datasets, whereas inventory-based approaches provide large but noisier samples. This tension remains a central challenge in developing reliable, scalable training data for lidar-based species recognition.

2.3.4. Botanical Taxonomy and Naming Standards

Consistent botanical taxonomy is crucial for resolving naming inconsistencies that arise in tree inventories and other training data sources. It provides the formal framework through which species labels can be harmonised across datasets and aggregated to broader taxonomic levels.

Botanical taxonomy classifies plants within a nested hierarchy of ranks: each individual belongs to a species, which is grouped into a genus, which in turn belongs to a family, and so on up to order, class, division, and kingdom. This hierarchical structure allows researchers to group taxa at different semantic levels depending on data availability and study objectives; for example, urban vegetation studies often classify trees at the genus or species level (Neyns and Canters, 2022).

Plant names follow Latin binomial nomenclature. The genus name is capitalised and the species name is lowercase, and both are italicised (e.g. *Quercus robur*). Cultivar names are written in single quotes after the species name (e.g. *Quercus robur* 'Fastigiata'), while hybrids are marked with a multiplication sign (×), placed either between the genus and species (*Platanus* × *hispanica*) or before the species name (*Populus* × *canadensis*). These conventions establish a universal standard for naming, enabling consistent communication and integration of datasets across studies.

This taxonomic framework underpins most vegetation-related remote sensing studies by providing a consistent way to label and compare tree observations across datasets. It is formalised in international nomenclature codes, such as APG IV The Angiosperm Phylogeny Group (2016), the International Code of Nomenclature for algae, fungi, and plants Turland et al. (2018), and the International Code of Nomenclature for Cultivated Plants (Brickell et al., 2016).

2.4. Urban Tree Reconstruction

Finally, this subsection reviews methods for reconstructing explicit tree geometries from point clouds for use in CFD. The reviewed studies highlight trade-offs between geometric realism, aerodynamic relevance, and computational efficiency.

2.4.1. Levels of Detail for Tree Models

Levels of Detail (LoD) frameworks define the granularity of tree models, clarifying how crown and trunk details can be progressively represented. LoDs are well established for buildings in 3D city models, most notably through the CityGML standard, which formalises LoD0 to LoD4 for both geometric and semantic representations (Gröger and Plümer, 2012).

In contrast, vegetation has lacked a comparable framework. To address this gap, Ortega-Córdova (2018) proposed a structured set of 14 refined LoDs for Single Vegetation Objects (SVOs), alongside four optional levels for root representation (Figure 2.1). These definitions capture geometric, semantic, and application-driven aspects of vegetation modelling, and are tailored specifically for urban contexts. This framework is adopted here as a reference for describing the level of detail achieved by the reconstructed tree geometries throughout this thesis.

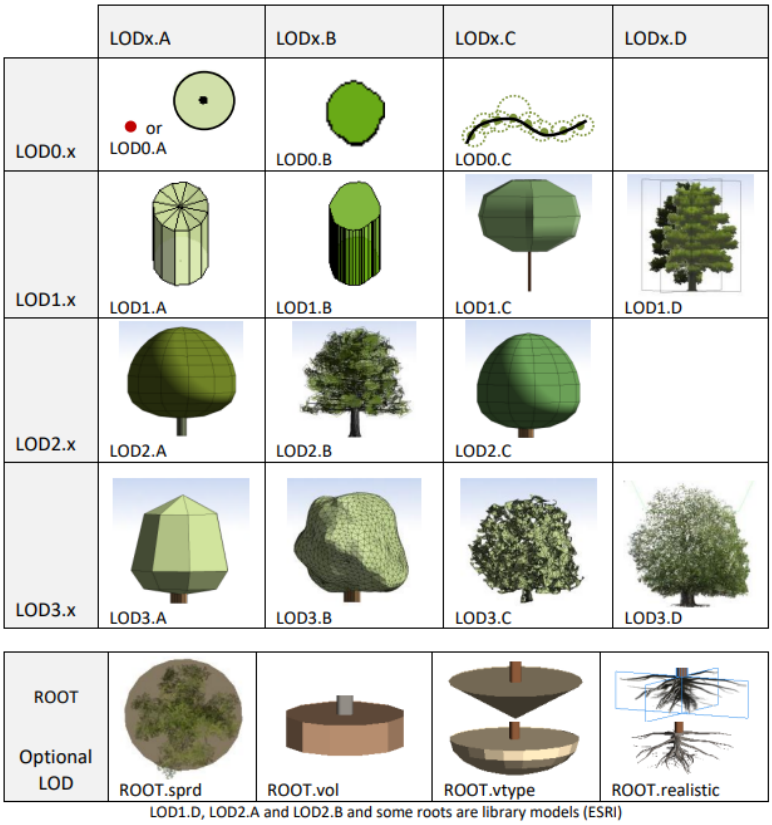
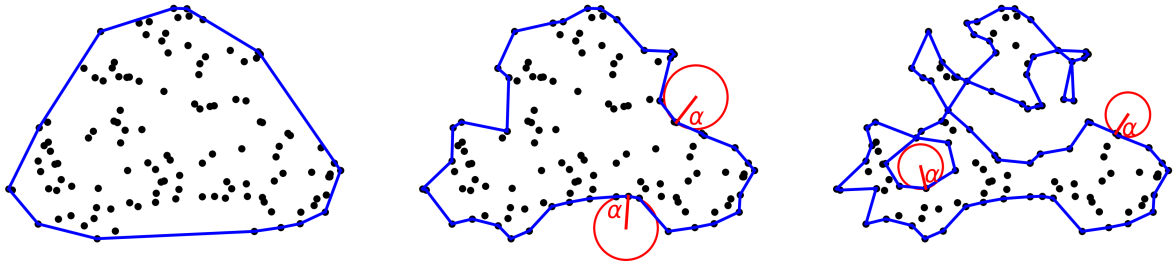


Figure 2.1: Levels of Detail for Single Vegetation Objects as proposed by Ortega-Córdova (2018).

2.4.2. Tree Crown Abstraction Methods from Point Clouds

Tree crowns can be reconstructed from point clouds using either surface-based or volumetric methods, depending on the desired level of geometric and aerodynamic detail. For CFD applications, these representations must provide watertight, non-intersecting volumes while remaining computationally manageable (see Section 2.1.5).

Alpha shapes, first formalised by Edelsbrunner and Mücke (1994), generalise the convex hull by allowing concavities controlled by a scale parameter α. Smaller α values yield more detailed, non-convex boundaries, while larger values approach a convex hull. The method forms a concave hull around a set of points using Delaunay triangulation, producing a polygonal surface that approximates the object envelope. de Groot (2020) applied this approach to delineate individual tree crowns from airborne lidar, generating smoothed crown volumes suitable for city-scale modelling. Such models align with LoD3 vegetation representation, where the overall crown form is required but internal branching is abstracted (Ortega-Córdova, 2018). A schematic comparison between a convex hull and α-shapes with varying scale parameters is shown in Figure 2.2, where decreasing α increases concavity but may also introduce non-manifold edges and disconnected components.



(a) Convex Hull (b) Alpha Shape (c) non-manifoldness

Figure 2.2: Comparison of geometric abstractions for a crown-like point set. (a) Convex hull enclosing all points. (b) α-shape with a moderate radius α, producing a connected concave outline; red circles illustrate the influence radius. (c) Smaller α results in fragmentation and non-manifold structures, highlighting the scale dependence of α-shapes.

However, classical alpha shapes may produce open or non-manifold surfaces when applied to sparse or noisy data. To address this, the *Alpha Wrapping* algorithm of CGAL extends the alpha-shape concept by enforcing manifoldness and watertightness (Alliez et al., 2025). De Groot's method employs classical alpha shapes, while this thesis adopts Alpha Wrapping to satisfy the CFD meshing requirements outlined in Section 2.1.5. Because García-Sánchez et al. (2021) demonstrated that geometric detail strongly influences wind-flow predictions, such watertight crown envelopes offer a practical balance between aerodynamic realism and computational efficiency.

An alternative approach to crown reconstruction is voxelisation, which converts a point cloud into a regularly spaced three-dimensional grid of volumetric cells (voxels). Each voxel represents a small cube of space that can be marked as either occupied or empty based on the presence of lidar points, resulting in a block-like volumetric approximation of the crown rather than an explicit surface mesh. An example of this process is illustrated in Figure 2.3.

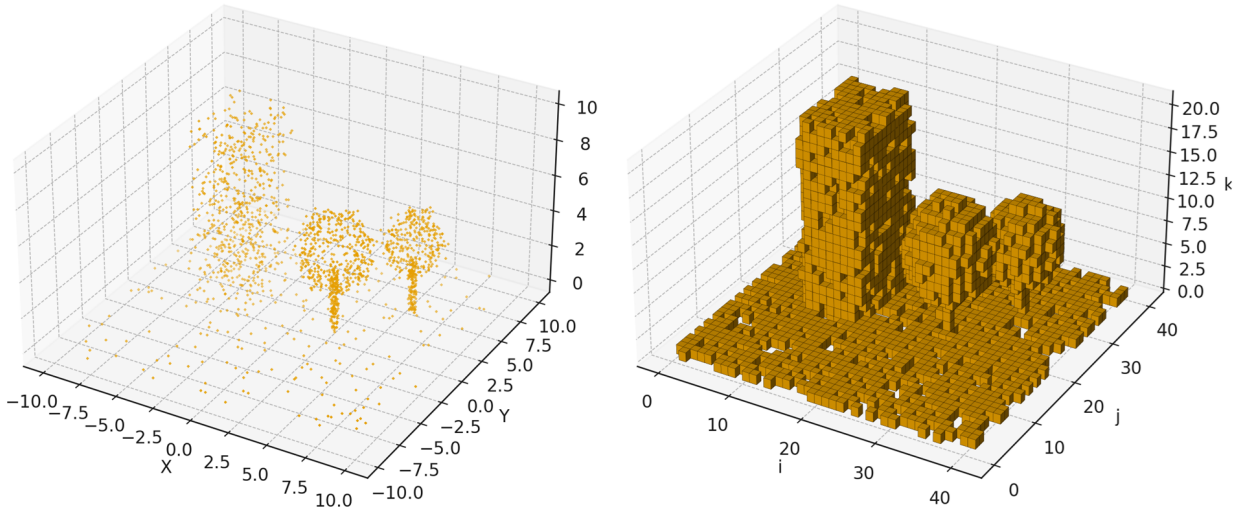


Figure 2.3: Example of voxelisation: (left) raw point cloud and (right) voxelised representation (0.5,m resolution). The voxelised model replaces the continuous crown surface with a regular volumetric grid suitable for porosity estimation.

Such voxel-based models describe canopy structure in a resolution-controlled way and enable the derivation of properties such as leaf area density (LAD) and porosity. Keerthinathan et al. (2025) showed that voxelised trees can provide sufficient structural detail for CFD studies focused on heat transfer, integrating naturally with grid-based solvers and avoiding the complexity of surface meshing. However, as García-Sánchez et al. (2021) note, excessive geometric simplification can distort airflow patterns, indicating that voxel grids may be too coarse for wind-focused CFD applications.

Overall, alpha-based and voxel-based methods offer complementary advantages. Voxelisation supports volumetric analysis, while alpha-based approaches provide explicit geometric boundaries required for CFD meshing. Given the need for watertight, low-complexity geometries emphasised in Section 2.1.5, Alpha Wrapping is adopted in this thesis as a CFD-ready solution for crown reconstruction, while voxelisation is used to derive porosity values.

2.4.3. Trunk Geometry Estimation from Crown Attributes

Trunk reconstruction approaches establish relationships between crown dimensions and stem geometry, enabling the generation of complete tree representations. While the crown exerts the dominant influence on aerodynamic behaviour, the trunk remains an important structural element that affects near-ground flow. Hong et al. (2017) demonstrated this in a RANS-based pollutant-dispersion study, where tree crowns were modelled as porous volumes and trunks as solid obstacles (see brief RANS definition in Section 2.1.5). Differentiating between these components substantially improved the accuracy of near-ground airflow predictions—an essential consideration for studies of pedestrian comfort and pollutant exposure.

Directly reconstructing trunks from airborne lidar (ALS) data remains challenging, as lower stems are often obscured by dense canopy layers. To overcome this limitation, Fu et al. (2020) developed a model that estimates diameter at breast height (DBH) from crown-level attributes such as tree height, crown area, and leaf area density (LAD), supplemented with species information. Their non-linear mixed-effects framework combines fixed effects (morphological traits) with random effects (e.g., microclimate or terrain variations), allowing for DBH estimation using aerial point-cloud data alone.

While Mao et al. (2023) highlight the inherent uncertainty of DBH prediction from airborne lidar—stemming from occlusion and variable point density—indirect approaches such as that of Fu et al. (2020) remain highly valuable. They offer a practical pathway for reconstructing trunk geometry from crown-derived metrics when ground-based measurements are unavailable, thereby improving the completeness of CFD-ready tree models.

2.5. Key Insights from Literature

The reviewed studies show that while individual components of tree modelling, such as crown segmentation, species classification, and geometric reconstruction, have been explored extensively, they are typically developed in isolation and for homogeneous forest settings. Urban contexts, with their irregular spacing, mixed species composition, and variable tree maturity, remain largely underrepresented. Moreover, previous work rarely links these components to the aerodynamic requirements of Computational Fluid Dynamics (CFD). Vegetation is often simplified to bulk porosity zones, and scalable methods to derive per-tree porosity from airborne lidar are still lacking. Although digital twins increasingly represent buildings in high detail, trees are often absent or heavily generalised. This lack of integrated, urban-focused methods frames the need for a unified workflow that connects segmentation, taxonomy-informed classification, and geometry reconstruction into CFD-ready models. The next chapter builds on this gap to define the central research question and outline the objectives of this thesis.

3. RESEARCH GOAL AND QUESTIONS

3.1. Research Gap and Goal

Most existing lidar-based vegetation studies focus on forested environments or small experimental plots. Urban settings, with their irregular spacing, heterogeneous crowns, and diverse age structure, remain underexplored. Moreover, few studies have explicitly linked tree segmentation and classification to the aerodynamic requirements of CFD. While digital-twin frameworks increasingly model buildings at high fidelity, trees are typically simplified or represented only as bulk porosity fields.

To address this gap, this thesis develops a scalable, automated pipeline that reconstructs CFD-ready models of urban trees from open-access airborne lidar data. The workflow integrates point-cloud segmentation, taxonomy-informed classification, and geometric reconstruction to produce simplified yet aerodynamically consistent tree representations. It is designed for city-scale application, preserving canopy and trunk structure while assigning per-tree attributes compatible with CFD simulations.

3.2. Research Questions

This brings us to the central research question of this thesis:

> *How can urban trees be automatically reconstructed from airborne lidar into CFD-ready models that capture canopy structure and drag characteristics while remaining scalable?*

This question is addressed through three sub-questions:

RQ1: *How can individual tree instances be segmented from an unstructured airborne-lidar point cloud in complex urban environments?*

RQ2: *Can urban trees be taxonomically classified from airborne lidar point clouds?*

RQ3: *How can segmented tree point clouds be simplified into CFD-compatible geometries that balance geometric abstraction with aerodynamic relevance?*

3.3. Scope and Assumptions

To ensure feasibility and reproducibility, the study is conducted under the following conditions:

- The study focuses on urban areas in the Netherlands, using AHN5 as the sole point-cloud source.
- Only trees are considered, although the workflow could in principle be extended to shrubs or other vegetation.
- The unstructured airborne-lidar point clouds are assumed to contain pulse-return information.
- No auxiliary imagery (e.g., orthophotos or satellite data) is required.
- Public municipal tree inventories provide taxonomic labels with assumed sufficient accuracy for training and evaluation.
- CFD-domain meshing, turbulence-model selection, and boundary-condition definition are beyond scope but follow standard CFD practice.
- The pipeline must remain computationally efficient at city scale, avoiding approaches that are highly accurate but non-scalable.

4. RESEARCH STRATEGY AND DATA SOURCES

4.1. Reader's Guide to the Workflow

This section serves as a reader's guide to the overall workflow used in this thesis. It outlines how the three research questions (RQs) are operationalised within a modular and scalable framework designed to reconstruct CFD-ready tree models from airborne lidar data (Figure 4.1). The pipeline consists of one preparatory stage followed by three analytical components, each addressing a specific research question and detailed in Sections 5–7.5. Together, these modules form a fully automated pipeline that converts unstructured airborne lidar data into CFD-compatible tree geometries.

- **Data Acquisition and Preprocessing (Sections 4.2–4.3).** Input datasets from multiple municipalities were standardised into a uniform format: (i) lidar point-cloud tiles clipped to municipal boundaries, and (ii) a point layer containing the location and taxonomic label of each public tree. This preparatory stage was deliberately designed to be lightweight, ensuring that subsequent processing steps can be executed efficiently at city scale—a key requirement for the scalability objective of this thesis.

- **Tree Extraction and Instance Segmentation (Chapter 5).** Vegetation points are isolated from the raw point cloud using the proposed *High-Order Multi-Echo Density (HOMED) vegetation filter*, which exploits the internal echo structure of lidar pulses to robustly distinguish vegetation from non-vegetation lidar returns. The resulting vegetation subset is segmented into individual tree clusters using the cuboid-based algorithm of Wang et al. (2018), tuned to AHN5 characteristics through custom performance metrics. This stage yields per-tree point clouds that form the basis for both taxonomic classification and geometric reconstruction, directly addressing **RQ1** on individual tree extraction from unstructured airborne lidar.

- **Taxonomic Classification (Chapter 6).** Each segmented tree cluster is linked to the corresponding municipal inventory record to evaluate whether structural and radiometric features enable *taxonomic classification* across multiple ranks (species, genus, family, and leaf type). Random Forest models are trained and assessed to examine the degree of separability achievable from airborne lidar alone. This component is exploratory and informs aerodynamic interpretation rather than constraining the geometric modelling process, thereby addressing **RQ2** on the feasibility of taxonomy inference from lidar-derived structure.

- **3D Geometry Reconstruction (Chapter 7).** Each segmented tree is reconstructed into a CFD-suitable model composed of a watertight, α-wrapped crown and a solid trunk. Crown porosity is estimated through adaptive voxelisation based on the characteristic gap size (d_{50}), yielding rotation-invariant volumetric porosity values within the CFD-consistent range (0.85–0.95). Trunk dimensions are derived allometrically from crown geometry, providing a simple cylindrical representation sufficient for aerodynamic drag modelling. The resulting trees are exported as `CityJSON` `SolitaryVegetationObjects`, each containing two geometries—a porous crown and a solid trunk—ready for use in CFD workflows and other digital-twin applications. This final stage directly answers **RQ3**, demonstrating how segmented tree point cloud clusters can be simplified into CFD-compatible geometries that preserve aerodynamic relevance.

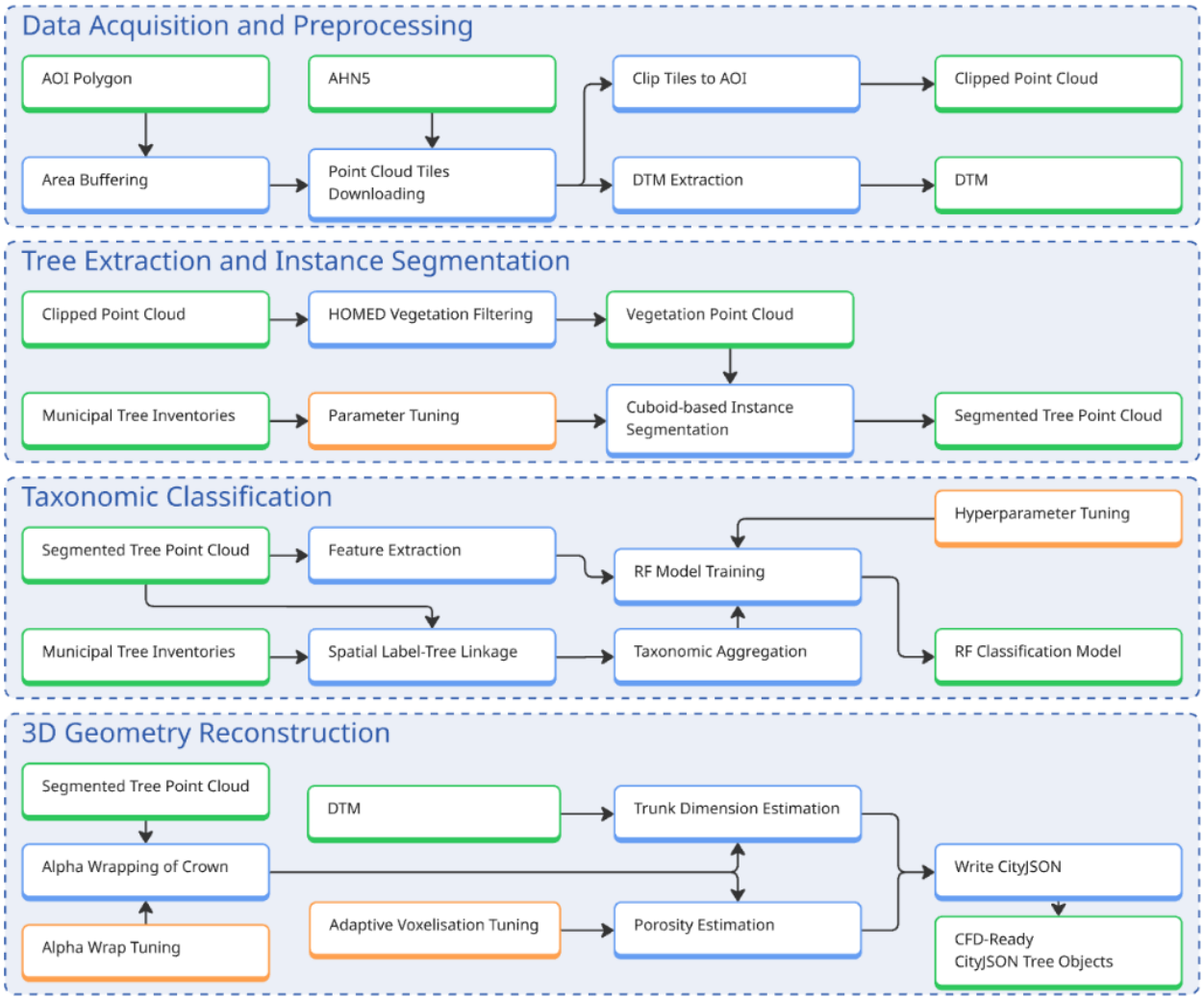


Figure 4.1: Overview of the complete workflow developed in this thesis. Blue boxes indicate algorithmic components, orange boxes denote parameter tuning or calibration stages, and green boxes represent major data outputs. The workflow operates fully automatically from raw airborne lidar to CFD-ready tree models and can be executed either as an integrated end-to-end pipeline or as individual modules. The taxonomy classification branch was explored separately and is not part of the final CFD reconstruction pipeline.

The workflow described above was implemented as an open-source software package to facilitate reproducibility and further development. All code developed for this thesis is available under a GPL-3.0 licence at `https://github.com/NoahAlting/CFTree`. The repository (`CFTree`) contains the complete implementation of the reconstruction pipeline, including modules for data acquisition, vegetation segmentation, and geometric reconstruction, as well as documentation and example configurations for reproducing the results presented in this thesis.

4.2. Point Cloud Data and Attribute Requirements

4.2.1. Data Source and Access

The point-cloud data used in this study originates from the **Actueel Hoogtebestand Nederland 5** (**AHN5**) dataset, the national airborne laser scanning (ALS) programme of the Netherlands maintained by the *Stuurgroep AHN*. AHN5 provides multi-return coverage of the entire country with

a nominal density of 10–20 points m^{-2} in urban areas and an accuracy of approximately 5 cm vertically and 10 cm horizontally. All data are referenced to the *Amersfoort / RD New* coordinate system (EPSG:28992) and fully documented in the official quality description (Stuurgroep AHN, 2023). AHN5 acquisitions are typically carried out under leaf-on conditions between late spring and early autumn, ensuring full canopy coverage but limiting the visibility of stems and ground surfaces beneath dense foliage. The acquisition dates vary locally between 2021 and 2024, as reported in the official AHN5 metadata.

To ensure manageable data volumes and enable parallel tile processing for scalable reconstruction, AHN5 tiles were accessed via the **TU Delft GeoTiles** platform (van Natijne et al., 2023), which distributes the national dataset as 1×1.25 km subtiles with 25 m overlap. Each GeoTiles tile retains all AHN attributes and additionally provides per-point RGB and NIR values derived from the national orthophoto mosaic.

Although AHN also publishes a national Digital Terrain Model (DTM), the DTM used for trunk anchoring (Section 7.4) was derived directly from the raw point cloud. This choice maintains a single-input workflow—reproducible entirely from the point-cloud tiles—while providing sufficient accuracy for geometric reconstruction.

4.2.2. Available Lidar Attributes and Their Roles in the Workflow

The AHN5 tiles follow the LAS 1.2 specification, providing geometric, radiometric, and semantic attributes per point. Table 4.1 summarises those relevant to this study. The reconstruction pipeline relies only on three attributes—(*X, Y, Z*) coordinates and the *Return Number*/*Number of Returns* pair—for vegetation identification using the *HOMED vegetation filter*. Taxonomy-based classification (Chapter 6) additionally requires the *intensity* attribute, following the feature set of Chi et al. (2025); the full list of predictors is given in Appendix C. All other LAS fields are optional or unused in this workflow.

Table 4.1: Lidar attributes included in the tiles that are accessible from Geotiles, along with their meaning and relevance to this thesis's pipeline.

LAS Field	**Meaning and Relevance**
X, Y, Z	3D coordinates of each lidar return; form the geometric foundation for all processing stages, including filtering, segmentation, and reconstruction.
Return Number / Number of Returns	Identify the order and total count of echoes per emitted pulse; used jointly in the *HOMED vegetation filter* to distinguish vegetation from non-vegetation lidar returns.
Intensity	Backscatter amplitude per return; used as an additional feature in the taxonomy-based classification (Chapter 6), but not required for geometry reconstruction.
Classification	AHN semantic label assigned per point; not used, since vegetation is merged into class 1 (*unclassified*) together with various non-vegetation objects.
RGB + NIR	Orthophoto-derived colour values provided by GeoTiles; initially considered for vegetation filtering and taxonomy classification but excluded due to temporal and geometric projection artefacts (Section 4.2.3).

4.2.3. Rejection of Orthophoto-Derived Colour Data

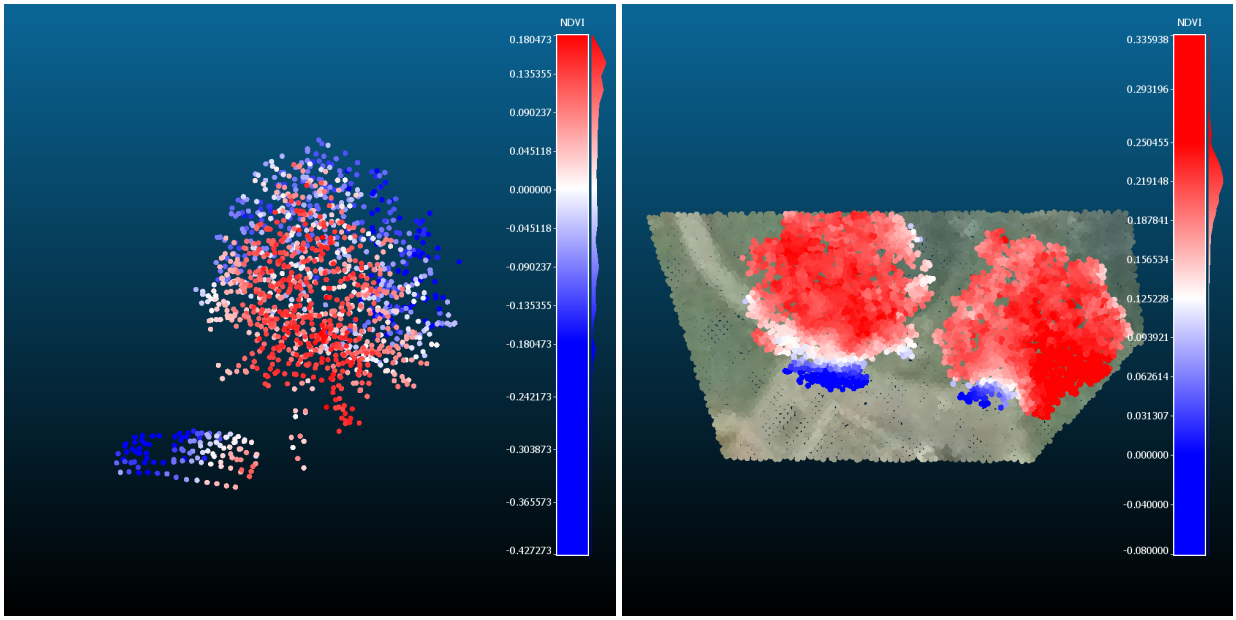
The RGB and NIR values included in the GeoTiles distribution are not native lidar measurements but colour attributes projected from 2D orthophotos. At the outset of this project, these bands

were used to compute the **Normalised Difference Vegetation Index (NDVI)**, defined as

$$\text{NDVI} = \frac{NIR - R}{NIR + R}, \tag{1}$$

to separate vegetation from non-vegetation points. NDVI is a well-established indicator of photosynthetic activity and would, in principle, be valuable both for vegetation filtering and for taxonomic classification based on foliage type (Haboudane et al., 2004).

However, the draped colour data were found to be unreliable. Two main sources of error were identified: (1) temporal differences between orthophoto and lidar acquisitions, causing objects such as cars or seasonal foliage to appear inconsistently, and (2) geometric mismatch from projecting 2D imagery onto 3D structures, where ground colours are mapped onto elevated points and vice versa. These effects produced false-positive NDVI responses over non-vegetated objects and false-negative responses within tree crowns (Figure 4.2). Consequently, RGB and NIR attributes were excluded from the pipeline, while lidar intensity—the only radiometric attribute directly measured by the sensor—was retained for use in the taxonomy classification experiments in Section 6.



(a) False-positive NDVI response on a parked car beneath tree canopy. Colour pixels from vegetation in the orthophoto were incorrectly mapped onto a non-vegetated surface.

(b) False-negative NDVI response in tree crowns due to orthophoto misalignment. Ground colours were projected onto elevated canopy points, underestimating vegetation indices.

Figure 4.2: Examples of NDVI artefacts caused by temporal and geometric mismatch between lidar geometry and orthophoto drape. Both cases illustrate why RGB and NIR attributes were excluded from the pipeline.

4.2.4. Applicability Across Lidar Modalities

The workflow is optimised for airborne lidar with moderate point density and multi-return capability, as provided by AHN5. Airborne sampling geometry leads to occlusion of lower canopy layers and stems, which limits the accuracy of fine-scale structural metrics but remains adequate for the aerodynamic abstractions targeted in this thesis. For terrestrial (TLS) or UAV-based lidar, the

increased ground-level detail and different sampling geometry would necessitate retuning parameters that depend on point density, such as the segmentation (Section 5.3.4) and alpha-wrapping settings (Section 7.2). The underlying principles of the *HOMED vegetation filter* (Section 5.2), volumetric porosity estimation, and geometric abstraction (Section 7.3-7.4) remain transferable to any dataset that records return-order information and captures sufficient canopy volume.

4.3. Municipal Tree Inventories and Preprocessing

Municipal tree inventories were used as ground-truth labels for training and evaluating the taxonomy-based classifier. These inventories record for every tree, among other attributes, a taxonomic name linked to a geographic location. The tree locations also supported validation of the instance-segmentation algorithm and the tuning of its parameters. Inventories were collected for Delft, Amsterdam, Rotterdam, and Utrecht (Table 4.2).

Table 4.2: Datasets used for this thesis, including scope, file format and data source.

Dataset	**Scope**	**format**	**Source**
Geotiles	NL	.laz	`geotiles.citg.tudelft.nl`
AHN Subtiles	NL	.shp	`weblog.fwrite.org/kaartbladen/#geotiles`
Boombeheer	Delft	.geojson	`data.delft.nl/datasets/d83a50486b384bfe8038c2d762f5e628_0/explore`
Stamgegevens	Amsterdam	.geojson	`data.amsterdam.nl/catalogus/bomen_stamgegevens_v1`
Bomen	Rotterdam	.shp	requested through email
Park en Straatbomen	Utrecht	.csv	`data.utrecht.nl/dataset/utrecht-geo-bomenkaart-update-2024`
Bomen in Bosplantsoen	Utrecht	.csv	`data.utrecht.nl/dataset/utrecht-geo-bomenkaart-update-2024`

Because the four cities used different conventions for taxon names and file formats, the inputs were standardised through a preprocessing step that produced two lightweight unified layers in the coordinate reference system `EPSG:28992` (Amersfoort / RD New): (i) a point layer with one record per tree containing only the coordinate and taxon, and (ii) a municipal boundary polygon used to clip the ALS point clouds.

To define a compact boundary enclosing the main tree areas, DBSCAN clustering was applied to the tree coordinates, followed by a two-dimensional alpha-shape envelope over the resulting clusters. This filtering step removes sparse or isolated records that would otherwise require downloading and processing additional point-cloud tiles with minimal analytical value. The resulting polygon was expanded by a 20 m buffer to ensure complete canopy coverage when clipping the point clouds. By limiting the data to the main connected canopy areas, the overall data footprint is substantially reduced, enabling the reconstruction workflow to scale efficiently to city-level datasets. Tiles intersecting each boundary were then retrieved from GeoTiles and clipped using PDAL. Figure 4.3 illustrates the derived boundary for Delft, and the corresponding polygons for Amsterdam, Rotterdam, and Utrecht are provided in Appendix A. Detailed statistics on the resulting sample counts and the number of retained training records after taxon aggregation are presented later in Section 6.2, where these quantities become directly relevant to the classification experiments.

Figure 4.3: Derived boundary polygon (blue) for Delft after applying DBSCAN clustering and a two-dimensional alpha shape to the municipal tree inventory (dots). Points outside the boundary (red) represent sparse or isolated trees excluded from the processing area to reduce the number and size of point-cloud tiles required for reconstruction, thereby enhancing the scalability of the pipeline.

5. TREE INSTANCE SEGMENTATION FOR POINT CLOUDS

5.1. Introduction and Motivation

Publicly available airborne laser scanning (ALS) datasets—such as the Dutch national elevation model (AHN5)—typically contain points representing laser returns from a wide range of surfaces and objects, including ground, buildings, vegetation, and other man-made or natural structures. However, these datasets often lack an explicit class label that distinguishes vegetation from non-vegetation points. Consequently, a dedicated filtering step is required to isolate vegetation points prior to any tree-level analysis.

In this study, vegetation is first separated from non-vegetation using a novel *High-Order Multi-Echo Density (HOMED) Vegetation Filter* (Section 5.2). This method was developed to address the absence of reliable vegetation labelling in AHN5 and other national ALS datasets.

After vegetation filtering, the resulting subset of points is segmented into per-tree clusters using *TreeSeparation*, a cuboid-based algorithm of Wang et al. (2018) (Section 5.3). Together, these two steps—*HOMED* filtering and *TreeSeparation*—transform raw ALS point clouds into individual tree-level point sets that provide the foundation for subsequent taxonomy classification and 3D geometry reconstruction.

5.2. High-Order Multi-Echo Density (HOMED) Vegetation Filter

5.2.1. Overview and Rationale

Before individual trees can be segmented, non-vegetation elements such as buildings, vehicles, and street furniture must be removed from the raw ALS point cloud. Including such points would confuse the segmentation algorithm, producing spurious clusters unrelated to trees. A dedicated vegetation filter is therefore required to isolate only those points that originate from vegetation.

Since publicly available ALS datasets, such as AHN5, generally do not provide a reliable vegetation class, a custom filtering approach—the *HOMED Vegetation Filter*—was developed. The filter exploits the physical behaviour of laser pulses interacting with foliage: when a pulse encounters volumetric vegetation, it typically produces multiple returns, and the early (non-last) echoes of such high-order pulses occur in dense spatial clusters. In contrast, non-vegetation features such as façades or vehicles yield single returns or sparse early echoes. By leveraging this difference in echo-order structure and local point density, the HOMED filter isolates vegetation points without relying on external classification labels, using only standard echo attributes available in modern ALS acquisitions.

5.2.2. Point Cloud Definitions and Notations

To describe the *HOMED* vegetation filter in a clear and reproducible manner, the key terms and operators are first introduced. These formal definitions (Table 5.1) establish the notation used throughout the method description, linking lidar pulse characteristics such as the *number of returns* (NR) and *return number* (RN) to derived sets of early echoes and their NR-based partitions. These partitions form the foundation of the HOMED filter, which identifies vegetation by analysing the density and spatial continuity of early high-order echoes.

Table 5.1: Definitions of symbols and point-set partitions used in the HOMED vegetation filter (symbolic; no numeric parameters).

Term / Operator	Mathematical	Explanation
Lidar return	$x \in X$	Each point or echo x with spatial coordinates and associated attributes.
Number of returns	$NR(x) \in \mathbb{N},\ NR(x) \geq 1$	Total echoes produced by the laser pulse that generated x.
Return number	$RN(x) \in \{1, \ldots, NR(x)\}$	Ordinal index of x within its pulse; the last echo of a pulse has $RN = NR(x)$.
Multi-return set	$X_{\geq 2} = \{x \in X \mid NR(x) \geq 2\}$	Returns from pulses that produced two or more echoes.
Early echoes	$E = \{x \in X_{\geq 2} \mid RN(x) < NR(x)\}$	Non-final echoes representing partial interceptions within volumetric targets.
NR-based partition	$E_n = \{\, x \in E \mid NR(x) = n \,\}$	Early echoes split into disjoint subsets according to the total number of returns per pulse. E_1 is undefined since single-return pulses are excluded from $X_{\geq 2}$.
Last echoes	$L = \{x \in X \mid RN(x) = NR(x)\}$	Final echoes per pulse, typically originating from ground or hard surfaces.

For convenience, the subset E_{4+} denotes the union of all higher-order early-echo partitions ($E_{4+} = \bigcup_{n \geq 4} E_n$). Figure 5.1 illustrates the echo-partitioning concept: laser pulses are grouped by total number of returns (NR), with early echoes ($RN < NR$) shown in blue (E_2), yellow (E_3), and orange (E_{4+}). Last echoes ($RN = NR$), typically from ground or other solid surfaces, are shown in red. Within the HOMED workflow, only E_3 and E_{4+} are retained, as E_2 lacks sufficient structural information to form a stable vegetation core.

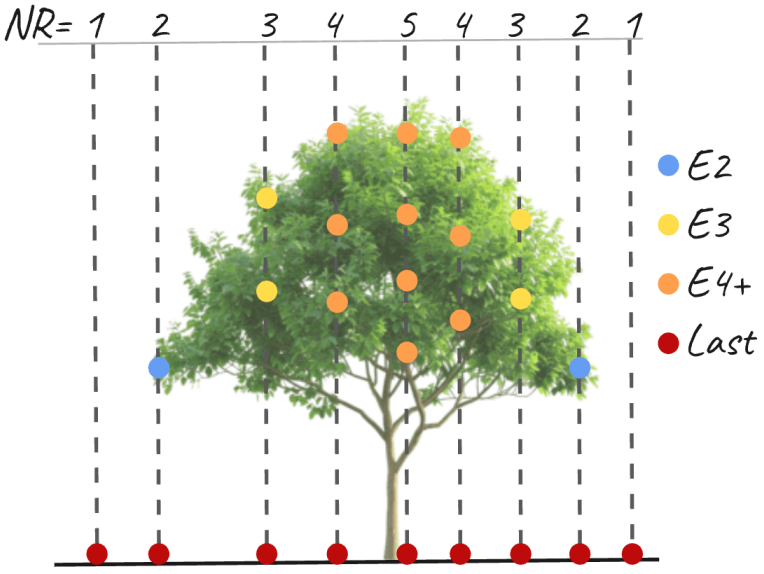


Figure 5.1: Schematic illustration of echo partitioning by number of returns (NR). Early echoes are grouped into subsets E_2, E_3, and E_{4+}, while last echoes ($RN = NR$) are shown in red. In practice, the HOMED filter retains only E_3 and E_{4+} for vegetation detection, as E_2 does not provide sufficient structural information to form a stable core.

5.2.3. HOMED Filtering Method

Building on the definitions in Table 5.1 and Figure 5.1, the *HOMED* vegetation filter applies the following sequential steps. Figure 5.2 provides an overview of the complete workflow, summarising the key filtering stages and their parameter settings. It illustrates how raw lidar input is progressively partitioned, denoised, and merged into vegetation subsets through echo partitioning, spatial outlier removal, pruning, and region growing. Each stage is described below, with parameter values explicitly reported to ensure reproducibility.

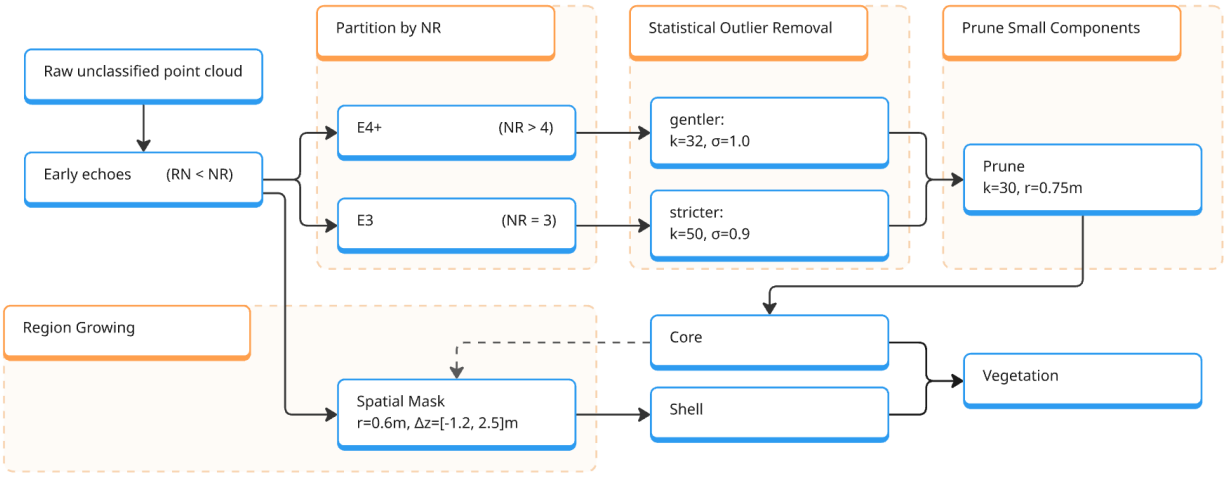
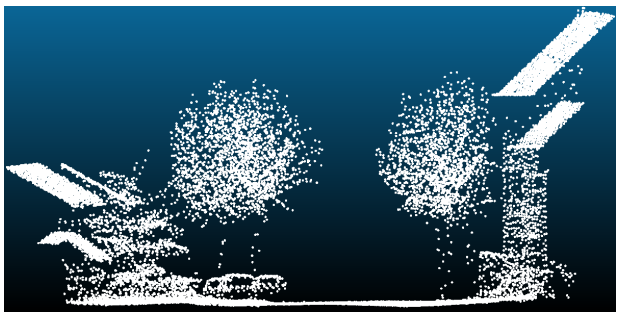


Figure 5.2: Schematic overview of the *HOMED* vegetation filtering workflow. The diagram shows the main processing stages, including echo partitioning based on the number of returns (NR), spatial outlier removal with two parameter sets (gentler and stricter), pruning of small clusters, and region growing to form the final vegetation subset. Parameter values are indicated to support reproducibility.

1. **Input.** The *HOMED* filter operates directly on the raw airborne laser scanning (ALS) point cloud, without relying on any external class labels, as outlined in Section 5.2.1. Figure 5.3 illustrates the type of input data used in this study. The scene represents a typical residential street in Delft and serves as an example for all filtering steps described in the subsequent sections.

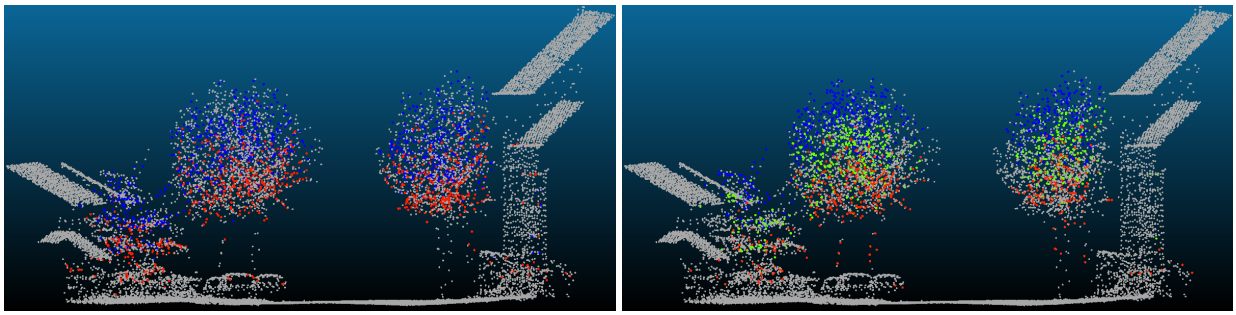
(a) Street-level view of the test area in Delft, showing dense roadside trees near buildings.



(b) Corresponding raw ALS point cloud containing both vegetation and non-vegetation returns.

Figure 5.3: Input data for the *HOMED* vegetation filter. (a) Google Street View image of the test area providing real-world context. (b) Raw ALS point cloud before filtering, containing mixed returns from both vegetation and non-vegetation. The *HOMED* filter operates directly on this raw input to identify and isolate vegetation points based on multi-echo density characteristics.

2. **High-Order Echo Partitioning.** Early echoes ($RN < NR$) are divided into two subsets, E_3 and E_{4+} (Figure 5.4). These correspond to all non-final returns from pulses that produced three or four-or-more total echoes, respectively. Although E_2 is defined (Table 5.1), this subset is excluded from further processing because it contains large amounts of non-vegetation noise (e.g., façades, vehicles, and reflective surfaces). Such points lack the volumetric structure required for stable vegetation detection, and their inclusion would prevent the subsequent statistical outlier removal from isolating a coherent tree cluster. In contrast, E_3 and particularly E_{4+} are characteristic of dense, multi-layered vegetation and therefore retained for the next filtering stage.



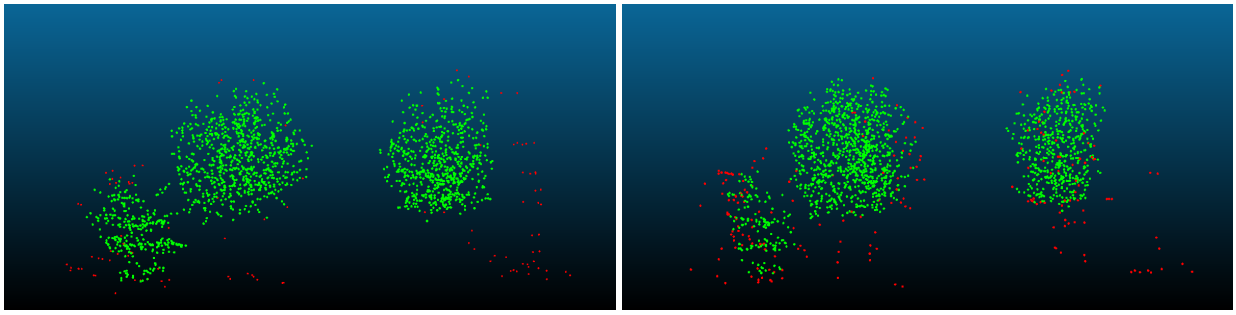
(a) E_3 (early only), colored by RN. (b) E_{4+} (early only), colored by RN.

Figure 5.4: Partition of early echoes by number of returns (NR).

3. **Statistical Outlier Removal (SOR) per partition.** For each subset, Statistical Outlier Removal is applied to eliminate isolated or spurious points (Figure 5.5). The filter computes the mean distance $\bar{d}(x)$ between each point and its k nearest neighbours; points whose distance exceeds $\mu + s \cdot \sigma$ are removed, where μ and σ denote the mean and standard deviation of all k-NN distances (Rusu et al., 2008). The parameter settings were determined empirically after testing several configurations:

$$E_3 : \; k = 50, \; \sigma = 0.9 \text{ (stricter)} \qquad E_{4+} : \; k = 32, \; \sigma = 1.0 \text{ (gentler).}$$

The stricter threshold for E_3 reflects that this subset may include ambiguous returns, for example from reflective façades or glass surfaces. In such cases the *HOMED* filter remains deliberately conservative, preferring to remove a few genuine canopy points rather than retain non-vegetation noise. For E_{4+}, where most points are expected to represent tree crowns, the threshold is gentler to preserve vegetation structure.



(a) SOR on E_3. Green retained, red removed. (b) SOR on E_{4+}. Green retained, red removed.

Figure 5.5: Partition-wise Statistical Outlier Removal (SOR). Each partition is filtered independently using empirically chosen parameters.

4. **Spatial Pruning of Connected Components** The retained points from both partitions are merged to form the candidate vegetation set, referred to as the *core*. Within this set, neighbouring points are connected into spatial components based on a fixed neighbourhood radius of 0.75 m. Each component represents a locally continuous cluster of points that is likely to correspond to a single piece of vegetation. To suppress spurious detections such as small noise patches or isolated returns, components containing fewer than 30 points are removed. This spatial pruning step therefore preserves only well-connected, volumetric regions that exhibit sufficient local point density to be interpreted as vegetation structure.

5. **Spatial growth of vegetation cores.** To recover vegetation points that were not captured in the core but are spatially connected to it, a surrounding *shell* is generated through two-dimensional (XY) dilation with a resolution of 0.25 m and a radius of 0.6 m (Figure 5.6). Vertical limits are set to $+2.5$ m above and -1.2 m below the nearest core point, and only early echoes ($RN < NR$) are admitted to the shell. Because the core has already been denoised by the preceding SOR stage, this spatial mask selectively includes additional canopy points without introducing significant non-vegetation noise. The parameter values were determined empirically through iterative visual inspection, seeking a balance between canopy completeness and noise suppression. This tuning is pragmatic rather than automatically optimised, reflecting adaptation to the characteristics of the input data.

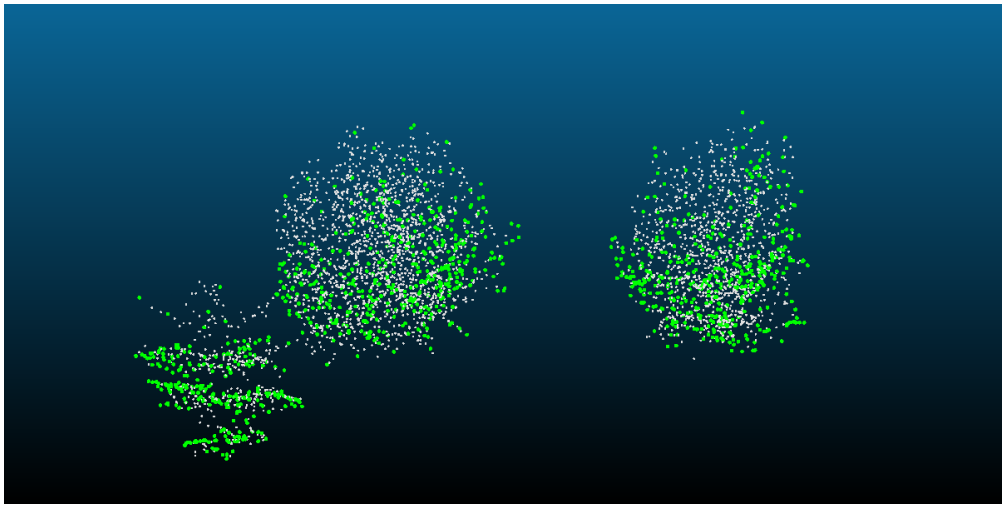


Figure 5.6: Core (white) and shell (green) produced by spatial growth in the *HOMED* filter. The shell extends the core to recover canopy regions that were partially missed during echo-based filtering.

6. **Final vegetation.** The final vegetation subset is defined as the union of the core and the shell. As illustrated in Figure 5.7, the *HOMED* filter effectively removes ground, building, and vehicle points, isolating individual tree crowns. A small fraction of canopy points is inevitably omitted, but this conservative bias is intentional: it is preferable to slightly under-represent the crown than to retain non-vegetation noise. Some peripheral crown points are also excluded, as these often correspond to first and only returns ($RN = NR$) that were not part of the candidate set during spatial growth in Step 5, which considered only early echoes. This effect is minor and acceptable for CFD purposes, as the majority of vegetation volume is retained. Tree trunks are also largely excluded, which is acceptable because they are already under-represented in airborne laser scanning data due to occlusion by foliage and view-angle effects. The HOMED filtering process may further reduce their presence, but this has limited impact on subsequent modelling, as trunk geometry is derived later from crown dimensions (see Section 7.4).

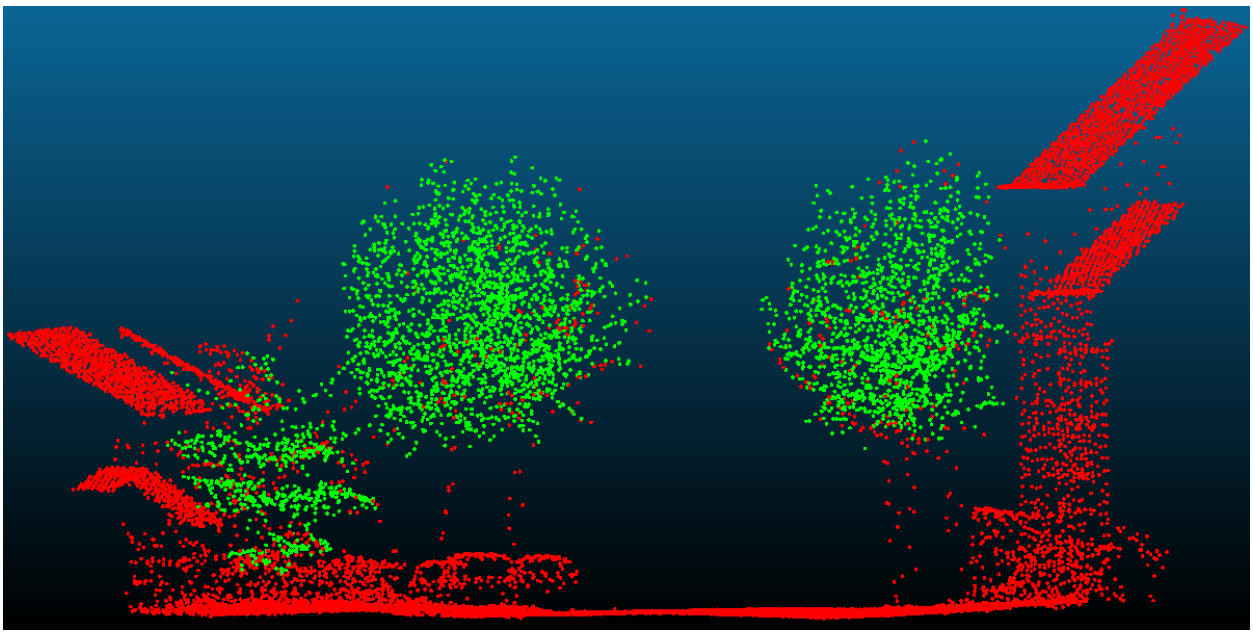


Figure 5.7: Final vegetation subset (green) compared to the raw point cloud (red). The *HOMED* filter removes ground, buildings, and vehicles while isolating tree crowns. Some peripheral canopy points are absent because first and only returns ($RN = NR$) were not included in the spatial growth step. Tree trunks are mostly missing due to occlusion and the conservative filtering strategy.

5.2.4. Results and Limitations

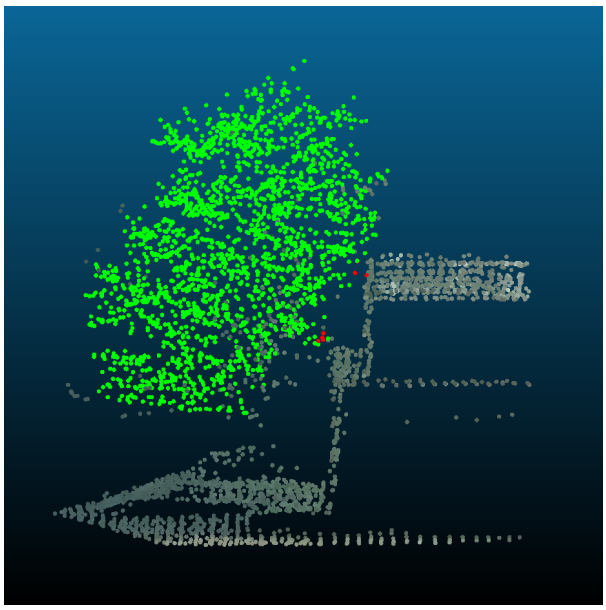
Although AHN5 includes a general class label (`class=1`, *unclassified*) that loosely contains vegetation, this category also includes numerous non-vegetation objects such as benches, traffic signs, and street furniture. To assess how the *HOMED* vegetation filter performs relative to AHN's labelling, a comparison was carried out in Delft Centrum using two configurations: one where the filter operated only on AHN's *unclassified* points, and one where it used all points in the raw dataset.

The results were nearly identical:

Raw point count:	950,369 points,
HOMED on AHN's *unclassified* points:	93,188 points,
HOMED on raw data:	93,322 points.

The difference of only 134 points (0.014%) indicates that very few points outside AHN's unclassified category are accepted by the *HOMED* filter, demonstrating that its vegetation filter is effectively independent of AHN classification. The small discrepancies correspond to AHN building-labelled points (`class=6`) that were retained by *HOMED*—non-vegetation returns likely originating from façade or roof features with multi-echo behaviour (Figure 5.8). Overall, these results confirm that the *HOMED* filter identifies vegetation solely based on multi-echo density properties, making it independent of external or dataset-specific labelling schemes. This enhances its generalisability: the method can be directly applied to any airborne laser scanning dataset that includes return-number information, without requiring pre-existing classification layers or national labelling standards.

(a) Street-level view of the test site showing the tree canopy in direct contact with the building façade.



(b) Comparison between *HOMED* on raw data (green and red) and *HOMED* on AHN's *unclassified* points (green only).

Figure 5.8: Influence of AHN classification on *HOMED* vegetation filtering. (a) Google Street View image showing how tree foliage overlaps the building façade, producing mixed echo returns. (b) Representative example from Delft Centrum comparing *HOMED* applied to raw data and to AHN's *unclassified* subset. The small differences between both configurations originate from façade points labelled as buildings (AHN `class=6`). No differences occur within vegetation, confirming that the filter's output is effectively independent of AHN labelling.

While the *HOMED* filter is designed to conservatively suppress non-vegetation, several limitations must be acknowledged. Visual inspection of multiple output areas showed that the filter consistently removed non-vegetation points, with no façade fragments, vehicles, or other spurious objects observed to pass. Very small or sparse trees may be omitted because their echo structure is insufficient to form a stable high-order core. In the context of CFD, however, this is acceptable: trees too small to be captured by the filter can be considered to have negligible aerodynamic influence at the neighbourhood scale.

Seasonal sensitivity is another limitation. Because AHN5 was acquired under leaf-on conditions, the HOMED filter performs optimally when canopy foliage is present and generates dense clusters of early echoes. In leaf-off conditions, particularly for deciduous trees, the reduced echo density would likely weaken or prevent the formation of high-order vegetation clusters, reducing detection performance. Consequently, the current implementation is most reliable for datasets acquired during the growing season and may require adjustment for winter or mixed-season acquisitions.

The only systematic artefact observed was caused by construction cranes (Figure 5.9), which occasionally produce dense clusters of high-order returns ($NR \geq 3$). Because these returns mimic the spatial pattern of foliage, cranes can be retained by the *HOMED* filter and may subsequently be segmented as tree crowns. Similar artefacts could theoretically occur for scaffolding, transmission towers, or highly reflective building façades containing large glass areas, although these were not encountered in the inspected tiles. Potential remedies include adapting the parameters of the SOR or Spatial Pruning thresholds to better target elongated or repetitive non-vegetation structures. A simple height-based post-filter could also remove such objects, but this approach

was avoided since it would risk excluding legitimate elevated vegetation—for instance, trees located on rooftop terraces or along steep terrain. Due to time constraints and these trade-offs, such refinements were not implemented in the current workflow.

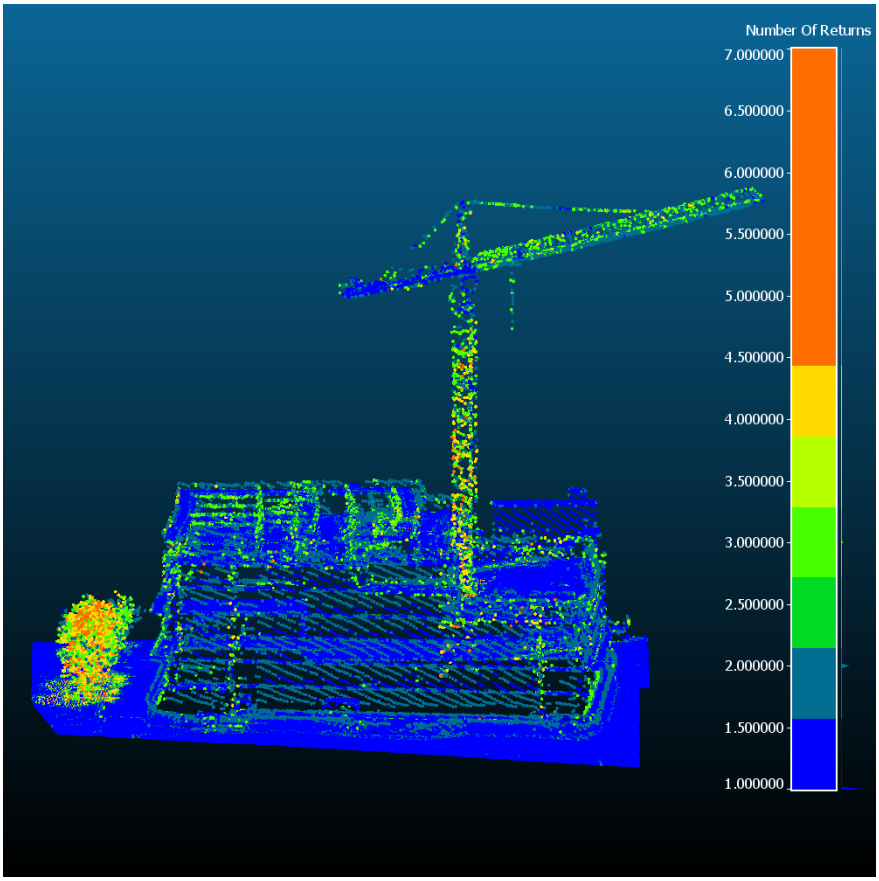


Figure 5.9: Example of a non-vegetation artefact: a construction crane producing clustered high-order returns ($NR \geq 3$). These points can pass the *HOMED* filter and may subsequently be segmented as tree crowns.

Despite these limitations, the vegetation subset produced by the *HOMED* filter provides a conservative and robust foundation for subsequent processing. In the following subsection, this vegetation point cloud is segmented into individual tree crowns using the method described in Section 5.3.

5.3. Tree Instance Segmentation

5.3.1. Overview and Motivation

Following the vegetation filtering step described in Section 5.2, the next stage separates the resulting vegetation points into individual tree clusters. This instance-level segmentation is essential for three reasons: it enables linking to municipal registries and associated attributes (e.g., species, height, canopy size); it provides the per-tree units required for constructing 3D crown and trunk geometries used in CFD simulations; and it facilitates consistency checks and quality control by comparing segmented clusters against registry data to identify potential filtering errors or spatial misalignments.

Segmentation in airborne lidar is challenging due to the heterogeneity of urban canopies. Crowns may be isolated, aligned in rows, or densely overlapping, while point density varies with flight parameters and occlusion. To address these conditions, the cuboid-based algorithm of Wang et al. (2018) (*TreeSeparation*) is adopted, as it combines scalability with robustness to canopy overlap and variable point density, providing suitable per-tree units for subsequent geometric reconstruction and aerodynamic parameterisation.

The remainder of this section describes the adapted segmentation algorithm and its top-down seeding variant for aerial data (Section 5.3.2), the evaluation metrics and indicators used for quantitative assessment (Section 5.3.3), and the selected parameter set applied in all city-scale experiments (Section 5.3.4).

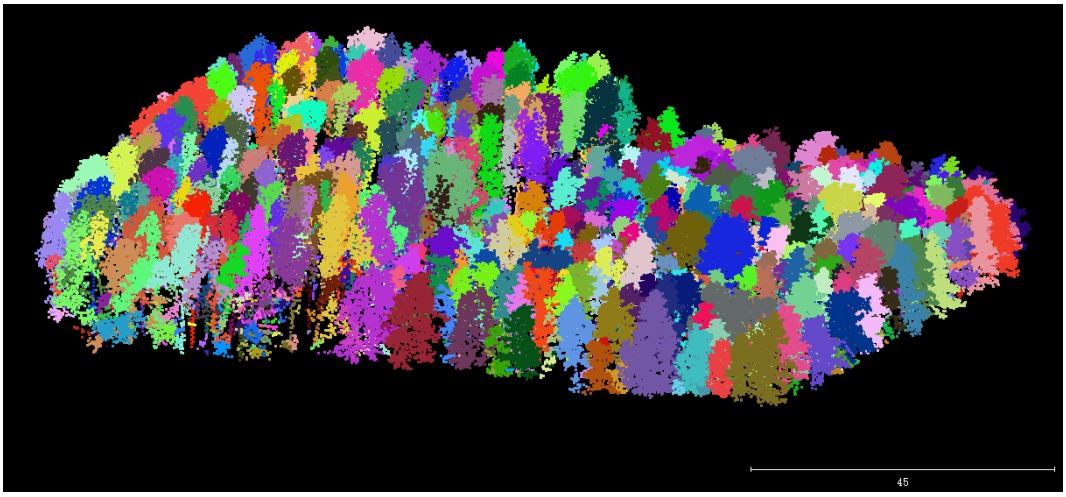


Figure 5.10: Example of tree instance segmentation on point cloud data. Each detected crown is assigned a distinct colour. The input is the vegetation point cloud obtained after filtering (Section 5.2). The output shows how crowns are separated into individual instances, which are later used for per-tree geometry reconstruction and CFD analysis.

5.3.2. Cuboid-Based Tree Instance Segmentation Algorithm (TreeSeparation)

The vegetation-only point cloud—extracted via echo-based filtering (Section 5.2)—is segmented using *TreeSeparation*, the cuboid-based algorithm proposed by Wang et al. (2018). The method partitions the point cloud into vertically stacked cuboids, identifies canopy seed regions, and propagates labels either downward (top–down) or upward (bottom–up) to form distinct tree instances (`tid`). Because airborne lidar often fails to capture the lower canopy and trunks, this work adopts the *top–down* variant, in which segmentation is seeded at the canopy level.

A central innovation of the method is the use of *cuboids* instead of uniform voxels. By stretching cells vertically (larger z) and refining them horizontally (smaller x, y), the algorithm achieves finer separation of adjacent crowns without a proportional increase in memory or processing load. After partitioning, cuboid cells are clustered via adjacency analysis. Tree-top candidate regions ($S_1, S_2, \ldots$) are detected as local maxima of point density and serve as initial seeds. Labels are then propagated throughout the cuboid structure (Figure 5.11).

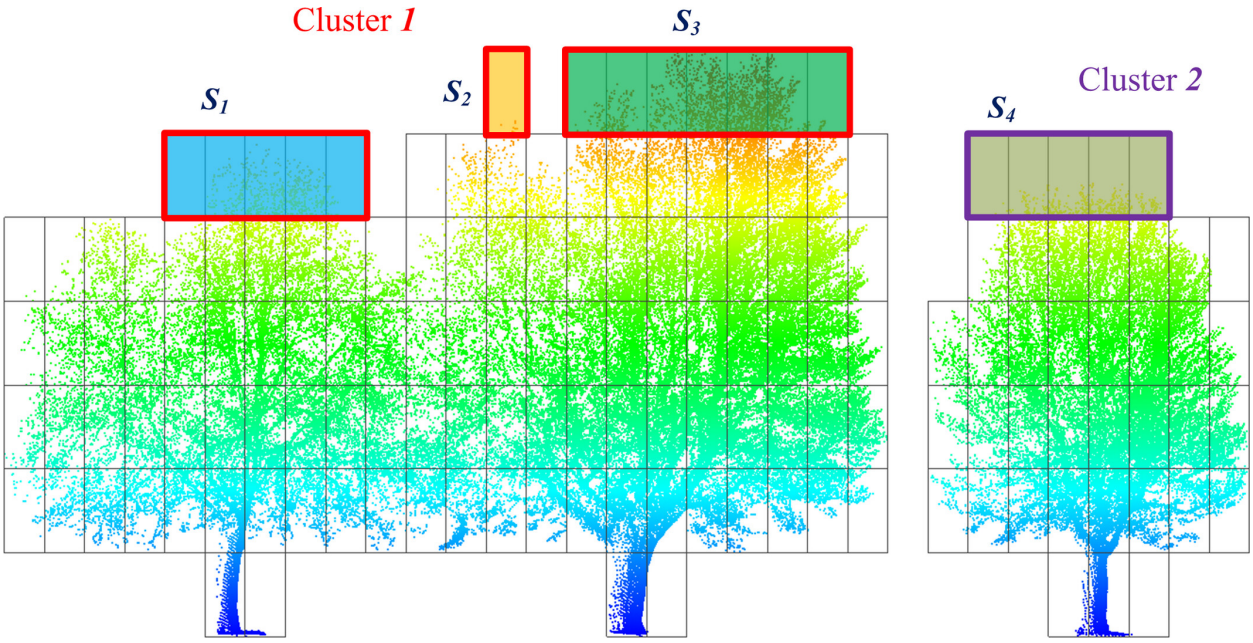


Figure 5.11: Schematic of *TreeSeparation* by Wang et al. (2018). The vegetation-only point cloud is partitioned into vertically stacked cuboids. Canopy seed regions (S_1–S_4) are detected at local maxima of point density and initiate downward label propagation. Adjacent crowns are thereby separated into distinct instances (Cluster 1 and Cluster 2).

The assignment of cuboids to a tree instance is not based purely on distance but also on a *connectivity coefficient* that accounts for both spatial proximity and the type of cell connection. Face-sharing neighbours are weighted more strongly than edge- or vertex-sharing neighbours, leading to more nuanced and accurate separation in dense canopies. This criterion enables the method to handle touching crowns and partial occlusions that are common in urban settings.

The algorithm features three interpretable parameters:

- **Radius** r [m] — lateral neighbourhood radius for merging cuboids that belong to the same tree;
- **Vertical resolution** v [m] — height of each cuboid layer along the z-axis;
- **Minimum points per cuboid** m [#] — occupancy threshold to consider a cuboid as part of a candidate tree structure.

These parameters have direct geometric meaning, making the method both efficient and scalable across point densities. However, given the heterogeneity and occlusions typical of urban canopies in AHN data—ranging from young planted trees to mature specimens—no single setting can be assumed optimal a priori. A systematic parameter search is therefore required to identify values that are robust across different neighbourhoods. To quantitatively define what constitutes a "good" segmentation, and to determine the optimal parameter triplet (r, v, m) for aerial lidar in urban areas, the following evaluation framework is introduced (Section 5.3.3).

During implementation, a minor bug was discovered that could cause the program to hang during the tree export stage. The issue originated from incorrect access to elements of the `std::map` container within the `outputTrees()` function, which assumed sequential integer keys. The iteration was revised to loop directly over key–value pairs, ensuring stable access to all tree instances. Additionally, the random RGB assignment was removed from the output routine, as colour information was not required for any downstream processing. A small functional extension was also introduced, allowing the compiled binary to accept command-line arguments specifying the input file, output file, and the three main segmentation parameters (r, v, m); this avoids recompilation when testing different configurations. These changes improved stability and usability without affecting the segmentation logic or results. Implementation details are provided in Appendix B.

5.3.3. Segmentation Evaluation Metrics

To assess segmentation quality, objective metrics are needed to quantify how well the instance clusters correspond to the municipal tree registry. In this context, a successful segmentation is defined as one that delineates individual crowns with minimal over- or undersegmentation, ensuring that each tree forms a distinct unit for subsequent CFD-compatible reconstruction. Let M be the set of municipality tree points within the area of interest (AOI), and H the set of 2D planar convex hulls obtained by projecting each 3D instance cluster onto the horizontal plane (x, y). For any municipality point $m \in M$ and any hull $h \in H$, define the incidence relations:

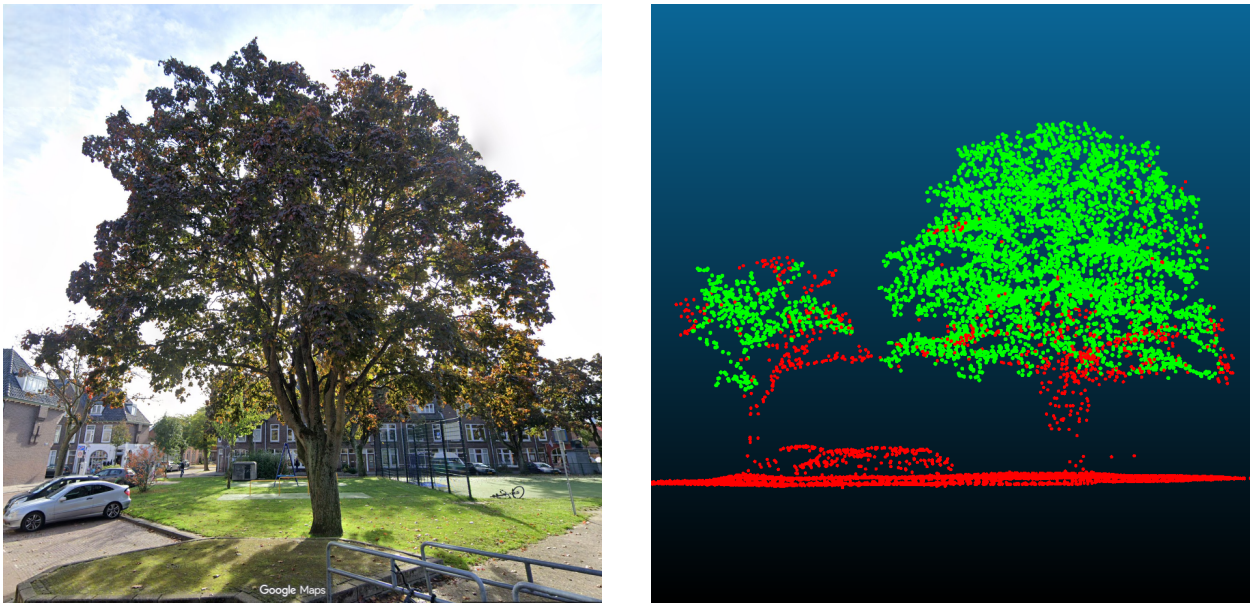
$$n_H(m) := |\{\, h \in H : m \in h \,\}| \quad \text{and} \quad n_M(h) := |\{\, m \in M : m \in h \,\}|.$$

Here, n_H counts how many hulls claim registry point m, while n_M counts how many registry points are covered by hull h.

From these incidence relations, the numerical metrics in Table 5.2 are derived and illustrated in Figure 5.12.

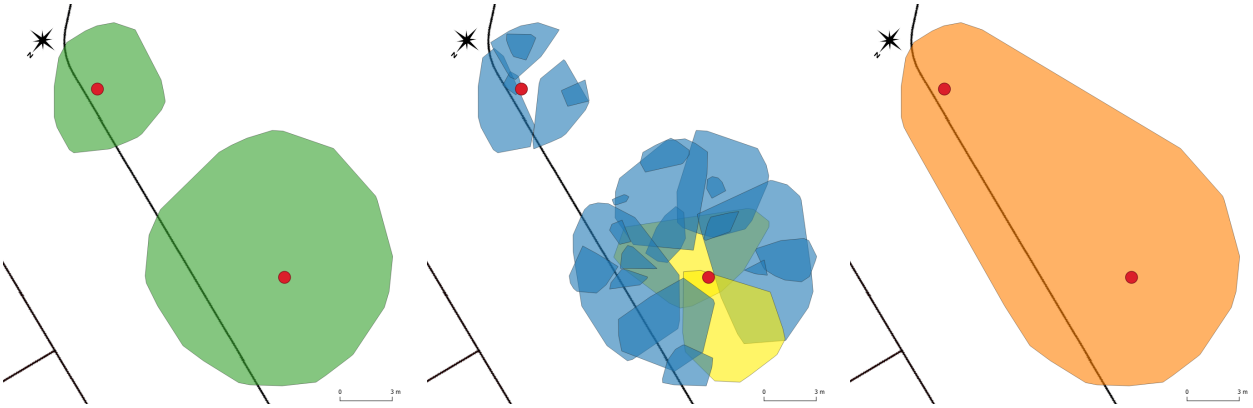
Table 5.2: Evaluation metrics for segmentation quality. Sets and functions are as defined in the text: M (municipal tree points within the AOI), H (segmentation hulls), $n_H(m)$ = number of hulls containing point m, $n_M(h)$ = number of municipal points inside hull h. The visual meaning of these metrics is illustrated in Figure 5.12.

Metric	**Symbol / Formula**	**Definition**
Muni total	$\lvert M \rvert$	Number of registry points after clipping to the AOI.
Points hit	$\lvert\{\, m \in M : n_H(m) \geq 1 \,\}\rvert$	Registry points covered by at least one hull.
Pure matches	$\lvert\{\, m \in M : n_H(m) = 1 \,\}\rvert$	True one-to-one matches (exactly one hull claims the point).
Oversegmentation points	$\lvert\{\, m \in M : n_H(m) \geq 2 \,\}\rvert$	One tree split across multiple hulls.
Undersegmented hulls	$\lvert\{\, h \in H : n_M(h) \geq 2 \,\}\rvert$	One hull that merges two or more registry trees.
Complexity	$\dfrac{\lvert H \rvert}{\max(1, \text{points\_hit})}$	Average hulls per hit registry point.



(a) Real-world tree (Google Street View)

(b) Point cloud after *HOMED* vegetation filtering



(c) Pure segmentation, $|H| = 2$

(d) Oversegmentation, $|H| = 19$

(e) Undersegmentation, $|H| = 1$

Figure 5.12: Example sequence illustrating segmentation evaluation context and outcomes. (a) Real-world photograph of the tree (Google Street View); (b) ALS point cloud after vegetation filtering — only green points are used for segmentation; (c–e) Comparison of segmentation outcomes between hulls (h) and municipal tree records (m). Red dots show registry points ($|M| = 2$); coloured polygons indicate segmentation hulls: green = pure one-to-one match ($n_H(m) = 1, n_M(h) = 1$); blue = unmatched hull ($n_M(h) = 0$); yellow = shared municipal point ($n_H(m) \geq 2$); orange = hull covering multiple municipal points ($n_M(h) \geq 2$). Complexity values ($|H|/|M|$) are 1.0 for (c), 19.0 for (d), and 0.5 for (e).

While these metrics describe a single segmentation result, they are not directly scale-free or comparable across AOIs. To allow fair comparison of parameter settings (r, v, m), four normalized indicators are defined (Table 5.3). These four indicators—P, OS, US, and C—capture complementary aspects of segmentation quality. High P indicates accurate one-to-one matches; low OS and US guard against splitting and merging errors; low C discourages excessive fragmentation. Using these scale-free indicators allows direct comparison across tiles and parameter settings without resorting to weighted composites.

Table 5.3: Scale-free indicators used for segmentation parameter selection. Here pure_H1_count, overseg_points, hulls_underseg, $|H|$, points_hit, and $|M|$ are defined in Table 5.2. The complexity is normalized per tile to be scale-free.

Indicator	Symbol / Formula	Meaning
Pure 1:1 rate	$P = \frac{\text{pure\_match\_count}}{\lvert M \rvert}$	Fraction of registry points with exactly one covering hull (true one-to-one).
Oversegmentation rate	$OS = \frac{\text{overseg\_points}}{\lvert M \rvert}$	Fraction of registry points claimed by two or more hulls.
Undersegmentation rate	$US = \frac{\text{underseg\_hulls}}{\max(1, \lvert H \rvert)}$	Fraction of hulls that contain two or more registry points.
Normalized Complexity	$C = \frac{\text{Complexity}}{\text{median}(\text{Complexity})}$	Hulls per "hit" registry point, normalized by the per-tile median.

Together, the metrics and indicators provide a transparent evaluation framework. The next subsection describes how they were applied in the parameter search to select a final (r, v, m) triplet.

5.3.4. Parameter Selection and Sensitivity

No parameter setting yields a segmentation that is free of both oversegmentation and undersegmentation while simultaneously achieving perfect one-to-one matches and minimal complexity. Instead, the aim is to find a balanced compromise: high P, while keeping OS, US, and C within acceptable ranges. To achieve this, a systematic grid search and sensitivity analysis was performed.

A Cartesian grid over (r, v, m) was explored (Table 5.4), applied identically to three representative Delft neighbourhoods—Centrum-Oost, Vogelbuurt, and Wippolder—that capture typical street-tree configurations with both dense pockets and sparser rows. Using these representative areas avoids overfitting to extreme cases while still exposing common failure modes. The grid comprised $9 \times 9 \times 4 = 324$ combinations per neighbourhood. Parameter ranges were initially adopted from Wang et al. (2018) and subsequently adapted to AHN data characteristics.

Table 5.4: Parameter grid explored for tree instance segmentation tuning. The grid spans a wide range of values for r, v, and m (324 combinations per neighbourhood), ensuring that both fine and coarse segmentations are captured. Applying the same grid to all three Delft AOIs allows direct comparability of parameter sensitivity across different urban contexts.

Parameter	Symbol	Values	Unit	Count
Radius	r	$1.0, 1.5, \ldots, 5.0$	m	9
Vertical resolution	v	$0.5, 1.0, \ldots, 4.5$	m	9
Minimum points per cuboid	m	$1, 3, 5, 8$	#	4

For each parameter value, the four indicators (P, OS, US, C) were summarised using the **median** and **IQR** across all settings of the other two parameters. This yields a 3×4 panel plot (Figure 5.13), where rows correspond to indicators and columns to parameters. Medians indicate the central trend, while IQR bands reflect sensitivity to the other two parameters. The visualisation makes trade-offs explicit—for example, increasing r steadily lowers OS and C but also reduces P and raises US.

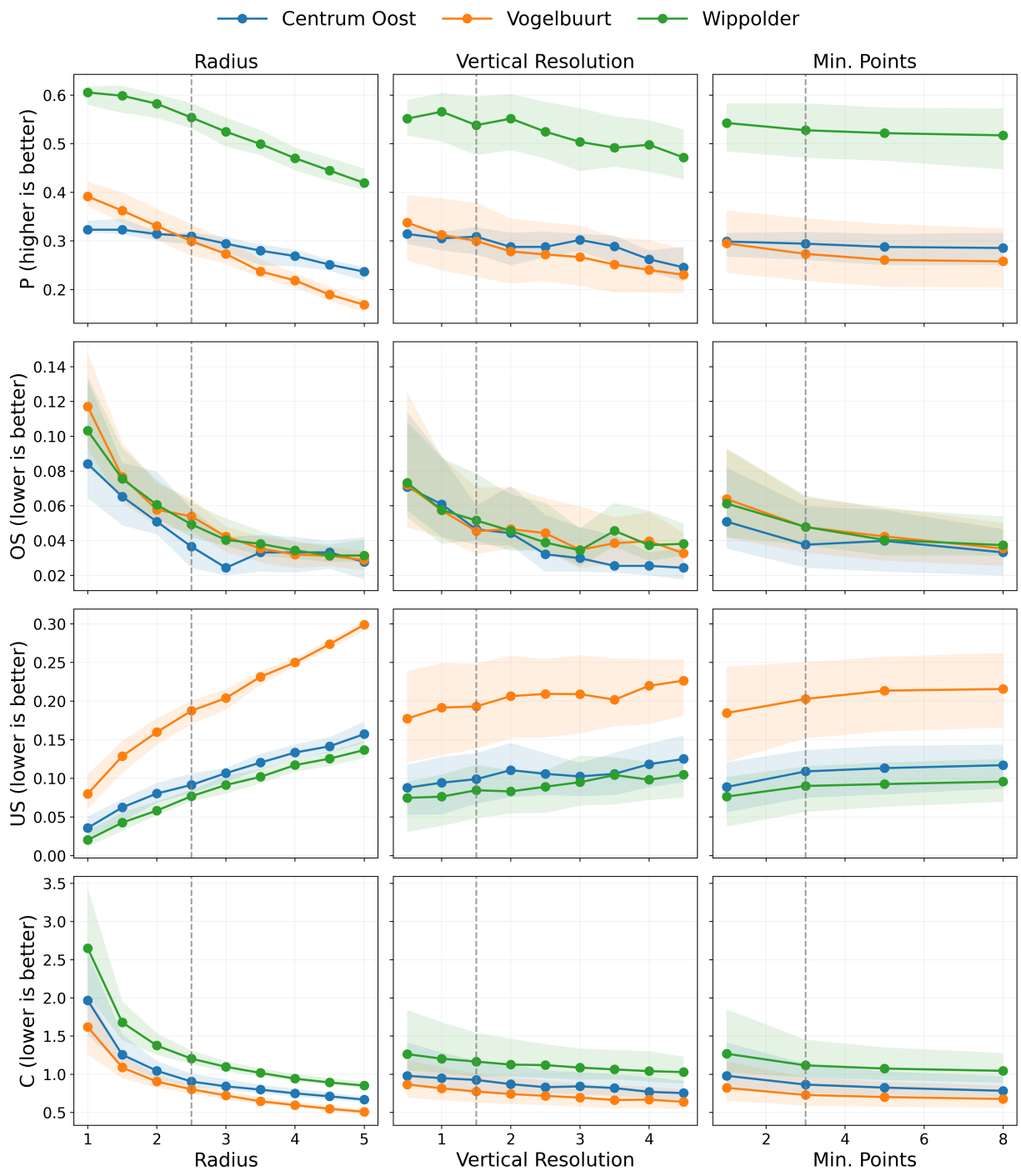


Figure 5.13: Sensitivity of segmentation quality indicators (P, OS, US, C) to the three parameters (r, v, m). Rows correspond to indicators and columns to parameters. Medians (solid lines) and interquartile ranges (shaded) are shown across the three representative AOIs. The plots highlight trade-offs: increasing r lowers OS and C but reduces P and raises US; coarser v decreases OS and C with minor P loss; larger m lowers OS but raises US, while P remains relatively stable. Dashed lines mark the final consensus setting $(r^*, v^*, m^*) = (2.5, 1.5, 3)$.

From these curves, a small and transparent selection rule was applied. First, the *smallest* radius r on the P-plateau was chosen, i.e. the smallest r such that $P(r) \geq (1-\varepsilon)\max_{r'} P(r')$. Beyond this point, larger r values reduce P without meaningful gains in OS or C. Next, the *coarsest* vertical resolution v that remains on the P-plateau was selected, keeping OS and C low without sacrificing P. Finally, an intermediate number of minimum points m was chosen to balance OS (which decreases with larger m) and US (which increases), noting that P is comparatively stable across m.

Applying this rule consistently across the three neighbourhoods yields a consensus setting[2]:

$$r^* = 2.5, \qquad v^* = 1.5, \qquad m^* = 3.$$

This triplet lies on the P-plateaus in all AOIs while maintaining low OS and normalized C and moderate US. The selected values are marked in Figure 5.13. Per-tile outcomes are reported in Table 5.5, and full hull maps for the chosen parameters are included in Appendix B for qualitative inspection.

Table 5.5: Per-tile segmentation results for the chosen parameters $(r^*, v^*, m^*) = (2.5, 1.5, 3)$. The triplet yields consistent performance across AOIs: P remains on the plateau, OS and normalized C are low, and US stays moderate. These values confirm that the selected setting balances the main trade-offs identified in Figure 5.13.

Tile	P	OS	US	C (raw)	C_{norm}	Figure
Centrum Oost	0.325	0.038	0.087	0.993	1.154	B.2
Vogelbuurt	0.333	0.058	0.172	0.860	1.183	B.3
Wippolder	0.574	0.063	0.072	1.294	1.169	B.4

5.3.5. Results and Limitations

Despite careful parameter tuning, residual segmentation errors remained. Touching or overlapping crowns were occasionally split into multiple clusters (oversegmentation), while sparsely sampled or small trees were sometimes merged into a single cluster (undersegmentation), especially where under-canopy occlusion reduced point density. Airborne laser scanning also rarely captures the lower canopy or trunk structure, which further constrains per-tree delineation. Examples of these segmentation errors are shown in Appendix B.

5.4. Interpretation and Discussion

This section reflects on the performance, limitations, and implications of the segmentation pipeline.

The vegetation filtering step proved crucial for removing non-vegetation objects before segmentation. An initial attempt combined the AHN classification labels with NDVI from orthophotos, but this approach proved unreliable due to artefacts from image draping. Instead, the *High-Order Multi-Echo Density (HOMED) vegetation filter* was developed. This method does not classify points directly; rather, it removes non-vegetation lidar returns to isolate a vegetation subset. The filter exploits the fact that lidar pulses interacting with foliage typically produce multiple echoes, analysing the spatial density of early high-order returns to identify volumetric vegetation. Combined with statistical outlier removal, this approach effectively suppresses sparse artefacts. The

[2]Other parameter combinations were also examined. While some gave broadly similar values of P, OS, US, and C, none offered a clearer overall balance. The chosen triplet therefore represents the most transparent and reproducible compromise under the plateau-based rule, avoiding the need for arbitrary weighting schemes.

resulting vegetation cores are then expanded with a spatial buffer to recover crown shell points, and clusters below a minimum size are pruned. Parameter choices for outlier removal, connected component pruning, and spatial masked region growing were determined empirically through visual inspection; in principle, these could be optimised further—for example, to better suppress occasional dense multi-return artefacts such as construction cranes.

Despite these limitations, the *HOMED* filter performed robustly in practice: very small trees were occasionally omitted, but their aerodynamic influence in CFD is likely negligible. Overall, the filter effectively isolates dominant canopy structures, ensuring that only vegetation with relevant aerodynamic impact is retained.

Segmentation of tree instances from the vegetation cloud using the *TreeSeparation* algorithm demonstrated a clear trade-off between robustness and sensitivity. Segmentation quality depended strongly on parameter choices. To formalise evaluation, four metrics were defined: one-to-one matches (P), oversegmentation (OS), undersegmentation (US), and complexity (C). These captured complementary error modes and allowed parameter sweeps to be assessed systematically. No parameter set achieved perfect results across all metrics, reflecting the inherent trade-offs in urban lidar tree segmentation. A plateau-based selection rule was therefore applied, yielding a consensus parameter triplet $(r^*, v^*, m^*) = (2.5, 1.5, 3)$.

The chosen parameters delivered stable and interpretable results across representative neighbourhoods, but residual errors remained. Oversegmentation split individual trees into multiple clusters, while undersegmentation merged neighbouring crowns. These errors propagate downstream: classification models may be trained on clusters that are not true single trees, and geometry reconstruction may either multiply or collapse tree instances. Despite these limitations, the segmentation step produced sufficiently consistent instances to support city-scale geometry generation.

These errors also influence downstream CFD performance. Oversegmentation yields multiple crown models per tree, each with slightly different porosity, but with gaps preserved between the segments, so aerodynamic consequences are expected to be limited. Undersegmentation, by contrast, merges multiple crowns into a single bulk canopy model, effectively filling the gaps between trees and likely overestimating wind resistance in those areas. In both cases, biases are systematic and occur at a scale that remains acceptable for neighbourhood-level simulations. This propagation of segmentation errors into aerodynamic outcomes underlines the need for accurate instance delineation, as established in Section 2.2, where segmentation was identified as a prerequisite for CFD-ready modelling.

A further implication concerns trunk estimation. In the current approach, trunk dimensions are inferred from canopy geometry (Section 7.4). Undersegmented clusters will produce too few trunks, while oversegmented clusters will produce too many. A dedicated treatment of trunk modelling is therefore required downstream.

Overall, the chosen parameter triplet represents a pragmatic balance: it produces consistent, interpretable tree instances across diverse urban settings, while maintaining errors at a level acceptable for downstream CFD analysis. These clusters form the input for subsequent classification and geometry generation steps described in Section 7.

5.5. Answer to RQ1 and Future Work

The preceding analyses and results provide a direct answer to the first research question:

> *How can individual tree instances be segmented from an unstructured airborne lidar point cloud in complex urban environments?*

The results show that urban trees can be reliably segmented from unstructured aerial point clouds using a two-step approach that combines the *High-Order Multi-Echo Density (HOMED) vegetation filter* with the scalable cuboid segmentation, *TreeSeparation*, adapted from Wang et al. (2018).

The *HOMED* vegetation filter identifies vegetation by analysing the density and spatial continuity of early high-order lidar echoes, exploiting the multi-return structure of pulses that interact with foliage. Sparse multi-return artefacts are removed using statistical outlier filtering, eliminating the dependency on AHN's vegetation class labels and ensuring dataset independence. Segmentation of individual trees is then achieved using *TreeSeparation*, tuned to AHN5 characteristics through custom performance metrics. Although oversegmentation and undersegmentation errors were unavoidable, a balanced parameter set was identified that minimised both, enabling consistent tree instances to be extracted at city scale.

Future work could focus on further reducing segmentation errors by refining how touching crowns are separated and how sparse under-canopy points are grouped, as well as by improving the HOMED filter's outlier and pruning parameters to better exclude rare non-vegetation artefacts such as cranes or scaffolding. Incorporating the upcoming AHN6 native colour data may also help recover small or sparse trees currently missed due to limited echo structure.

6. TAXONOMY-BASED TREE POINT CLOUD CLASSIFICATION

6.1. Introduction and Motivation

This section investigates whether individual tree instances segmented from AHN point clouds can be classified into taxonomic groups based on their geometric and radiometric characteristics. The motivation for this experiment was to explore whether such classifications could enable the assignment of typical aerodynamic parameters, such as crown porosity, on a taxonomic level rather than per individual tree. This would make large-scale urban simulations more scalable, especially in regions where detailed lidar data are unavailable but species inventories are.

Including trees in urban simulations is still uncommon, and most digital twin frameworks focus almost exclusively on buildings. This raised the question of whether taxonomic information could be extracted from lidar point clouds to complement geometric reconstruction. While geometric reconstruction can produce explicit 3D models, taxonomy-based classifications could allow parameterisation of trees based on expected physical traits, such as leaf persistence or branching density, which influence their aerodynamic behaviour.

Previous work has shown that species-level classification from point clouds is possible when combining geometric and spectral features: for instance, Chi et al. (2025) reported high accuracies using lidar-derived structural metrics together with hyperspectral data. This suggested that similar features might also work in an urban context using AHN data, provided that enough training samples were available.

Because of the time constraints of this project, the classification task was scoped to traditional machine learning using Random Forest, which is widely used in related work on vegetation classification (Chehreh et al., 2023). Random Forest is known to perform well with correlated features and is relatively fast to train. Convolutional Neural Networks (CNNs) have achieved higher accuracy in other studies (Hell et al., 2022; Chen et al., 2021a), but could not be implemented within the available time frame. The goal was therefore not to build a state-of-the-art classifier, but to determine whether taxonomic classification is feasible with this dataset and feature set.

Ultimately, the classification results did not reach a level of accuracy suitable for use in the reconstruction pipeline. The work is still included here because it represents an investigated path, and its outcome informed the final design of the pipeline: rather than assigning porosity per taxon, porosity is now estimated empirically for each tree instance.

6.2. Data Preparation and Taxonomic Aggregation

The classification experiments used tree instances segmented from AHN point clouds. Delft was initially used as a testing area, but its tree inventory contained too few samples to populate the taxonomic classes sufficiently. To obtain a large and diverse training dataset without the need for manual labelling, the tree inventories of Amsterdam, Rotterdam, and Utrecht were included. Each segmented tree crown was linked to the corresponding tree in the municipal inventory using a 2D convex hull–based spatial intersection, following the inventory-linkage strategies of de Groot (2020). A tree was considered linked if its inventory point fell within the projected crown hull.

However, segmentation errors can distort this relationship: oversegmentation can fragment a single crown into several smaller hulls, while undersegmentation can merge multiple trees into one. Hulls without any municipal match typically correspond to private trees or segmentation artefacts. To ensure clear correspondence between geometry and label, only **pure** one-to-one matches—where a single hull overlaps exactly one municipal record—were retained for training.

Figure 6.1 illustrates how the municipal tree points and segmented crown polygons are linked, including the colour coding used to distinguish the different cases. The overall linkage statistics for all cities are presented in Table 6.1, where these same categories—pure (green), oversegmented (orange), undersegmented (yellow), and unmatched (blue)—are summarised numerically.

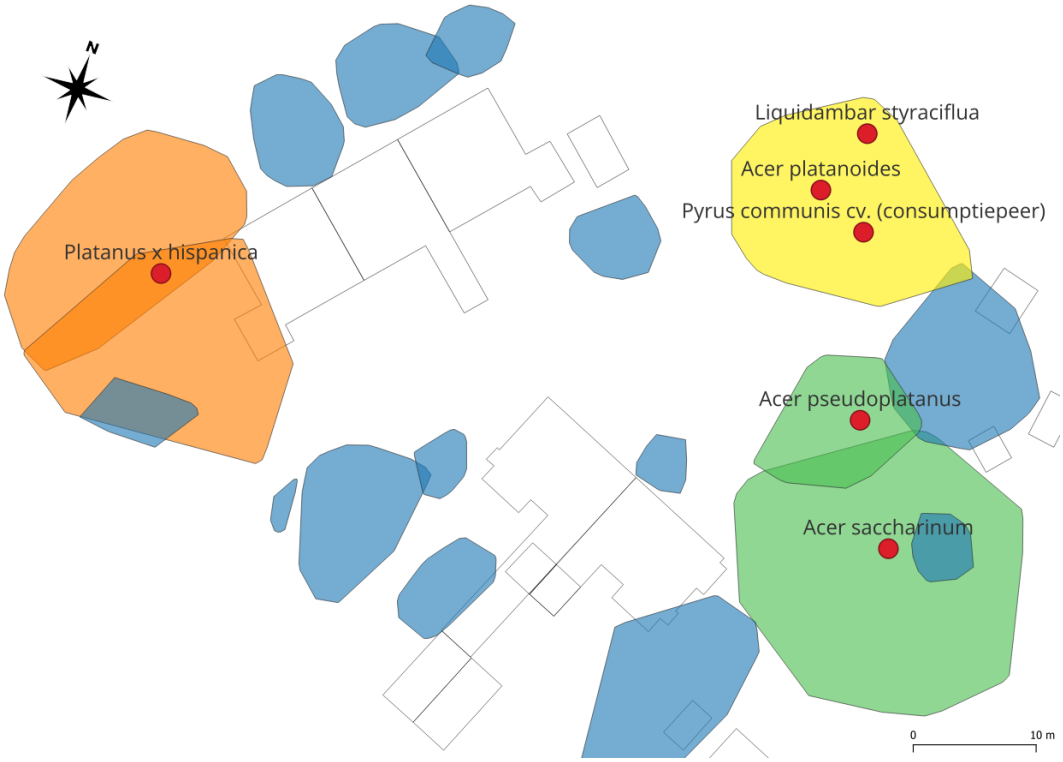


Figure 6.1: Linkage between segmented crowns and municipal inventory taxonomy labels. Red dots represent inventory tree records; green = pure one-to-one match, orange = oversegmentation, yellow = undersegmentation, blue = unmatched hulls. Only pure matches are used for training.

Table 6.1: Linkage statistics between municipal tree inventories and segmented crowns. Colour categories correspond to those shown in Figure 6.1: pure one-to-one matches (green) were retained for training, while oversegmented (orange), undersegmented (yellow), and unmatched (blue) hulls were excluded.

City	Tree Records	Total Hulls	Pure	Underseg.	Overseg.	Unmatched
Delft	36,367	50,815	14,441	2,694	5,521	28,159
Amsterdam	318,882	380,136	111,169	22,795	39,098	207,074
Rotterdam	204,967	291,878	87,386	15,035	20,455	169,002
Utrecht	156,898	182,650	62,592	10,020	18,274	91,764

Although a substantial share of tree records could not be linked one-to-one with segmented crowns, a sufficient number of pure matches remained for training. Across all four cities, **275,588 taxon labels** were obtained, representing approximately **31%** of all tree records. The high number of unmatched hulls is expected, as many correspond to private trees not included in municipal inventories or to segmentation artefacts. Despite these losses, the combined dataset still provides a large and diverse set of reliably labelled tree instances for training.

Because the labels originate from municipal inventories rather than field surveys, they contain some inconsistencies (e.g., cultivars or placeholder entries such as “species” or “to be classified”). In addition, the automatic labelling inherits errors from segmentation: oversegmentation and undersegmentation can respectively split or merge trees, leading to potential mislabelling. Although this approach was considered sufficiently accurate for large-scale training, it should be noted that the dataset is not manually verified by taxonomic experts.

The raw species-level label distribution was highly imbalanced: a few species dominated, while many others occurred only a handful of times. This imbalance is likely a result of municipal planting preferences, where certain species are planted frequently while others are rare, and possibly the wide range of tree ages present in urban environments (in contrast to forests), which makes individuals of the same species vary strongly in appearance.

Figure 6.2 shows the species label distribution in the training dataset. The species labels exhibit pronounced long-tail behaviour in which a small number of classes account for a large share of observations.

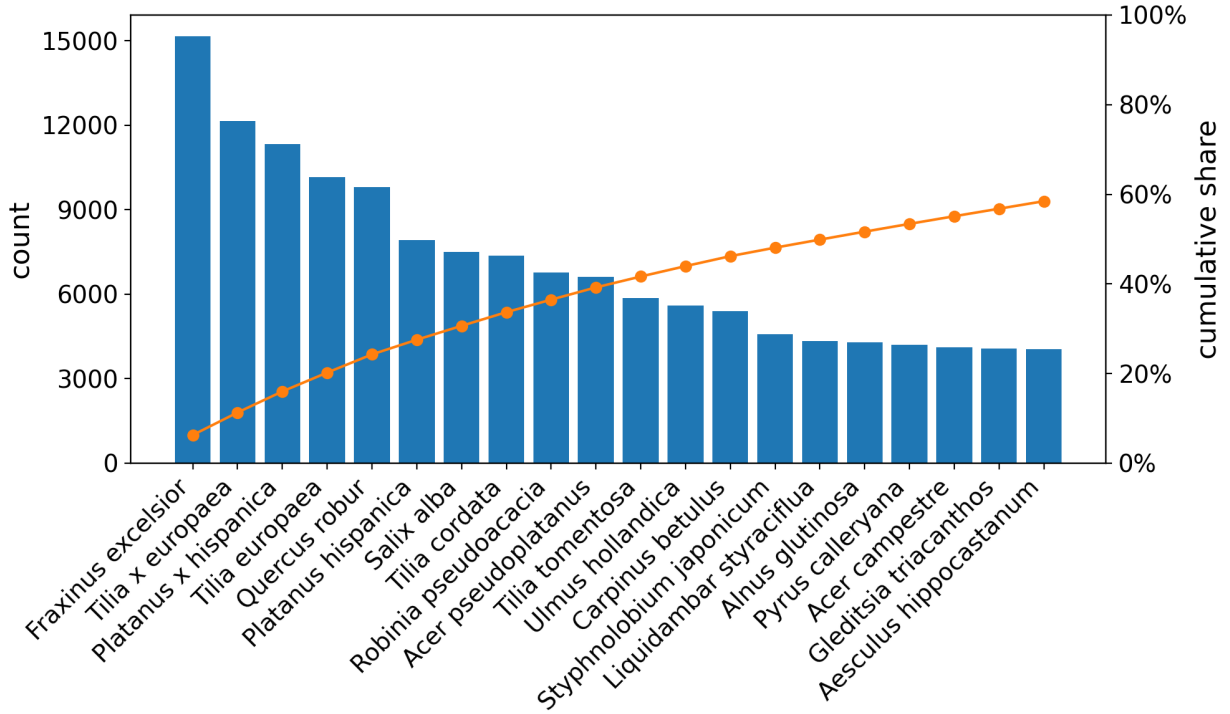


Figure 6.2: Class distribution (top 20) of species labels. Bars show sample counts, and the line shows cumulative share. The distribution is strongly long-tailed at the species level; of the total 472 unique species labels, the 20 most frequent classes account for 58.5% of the total samples. There were 275,588 samples in the training area that have a valid species label.

To train a classification model effectively, it is beneficial to have classes that occur with similar frequencies. The observed imbalance in the species domain substantiates the search for a label distribution that has fewer classes that account for a larger fraction of the training samples.

Municipal tree inventories record botanical taxa with varying levels of detail, ranging from cultivars to species or having Dutch tree synonyms rather than having scientific names. To reduce class imbalance and to report results at multiple levels of semantic detail, these labels were normalised and aggregated along the standard taxonomic ranks: cultivar $\rightarrow$ species $\rightarrow$ genus $\rightarrow$ family. A brief overview of botanical taxonomy and naming conventions is provided in Section 2.3.4.

For urban CFD modelling, the key distinction is whether trees retain their foliage in winter. To capture this, an additional binary label was introduced: *evergreen* or *deciduous*. These terms are used in the **phenological sense**: evergreen corresponds to "winter-on" (foliage retained through winter) and deciduous to "winter-off" (foliage shed in winter), rather than as synonyms for conifer or broadleaf. Labels are derived from the recorded family/genus/species name using a small list of overrides: if a species rule exists, it overrides the genus, which overrides the family. In practice, conifers are treated as evergreen except for well-known deciduous conifers such as *Larix*, *Metasequoia*, and *Taxodium*, while some broadleaf species are marked as evergreen (e.g. *Quercus ilex*, *Ilex aquifolium*, *Laurus nobilis*, *Magnolia grandiflora*). The full mapping is provided in Appendix D.

Figure 6.3 illustrates how the labels were aggregated from species to higher taxonomic ranks, and how the phenological leaf-type class was derived.

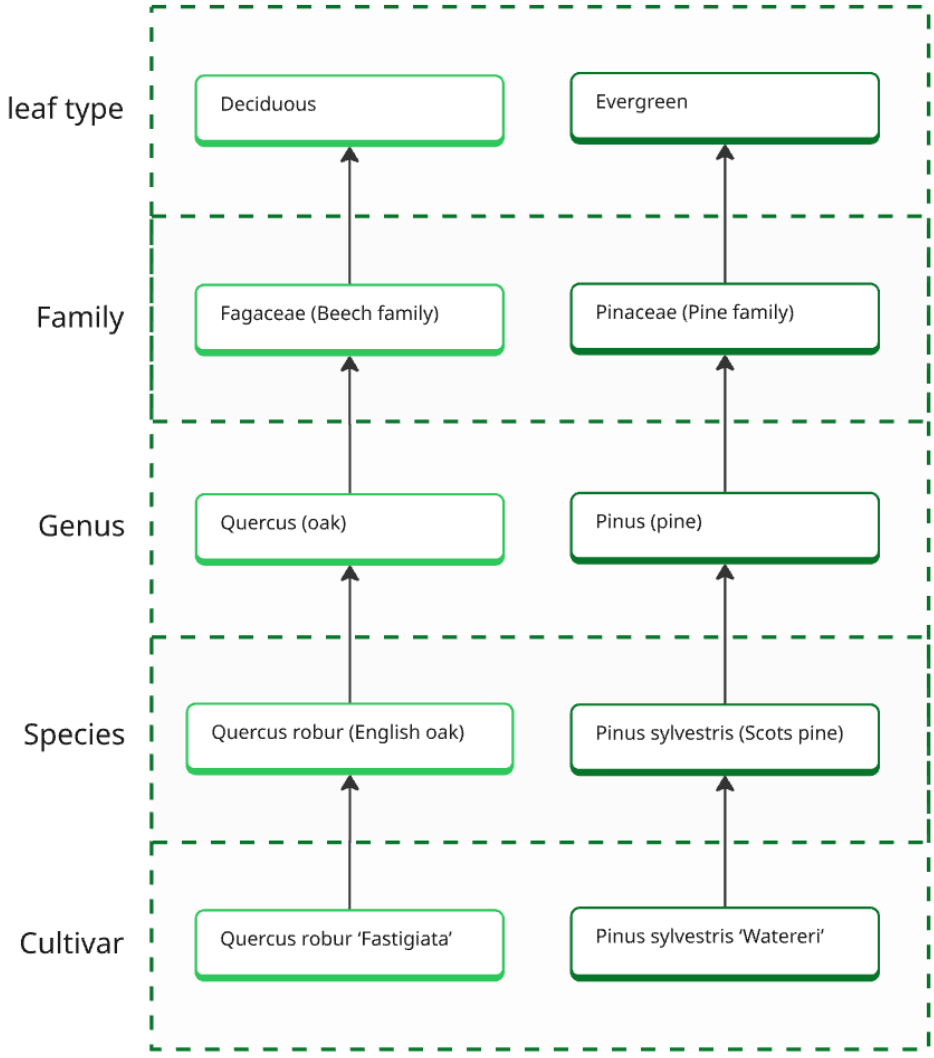


Figure 6.3: Illustration of the label aggregation by taxonomic relations: cultivar → species → genus → family, shown for one deciduous angiosperm lineage (left) and one conifer gymnosperm lineage (right).

Aggregating the tree taxa balanced the label distribution, allowing a larger share of samples to be represented by fewer classes. This effect is illustrated in Figure 6.4.

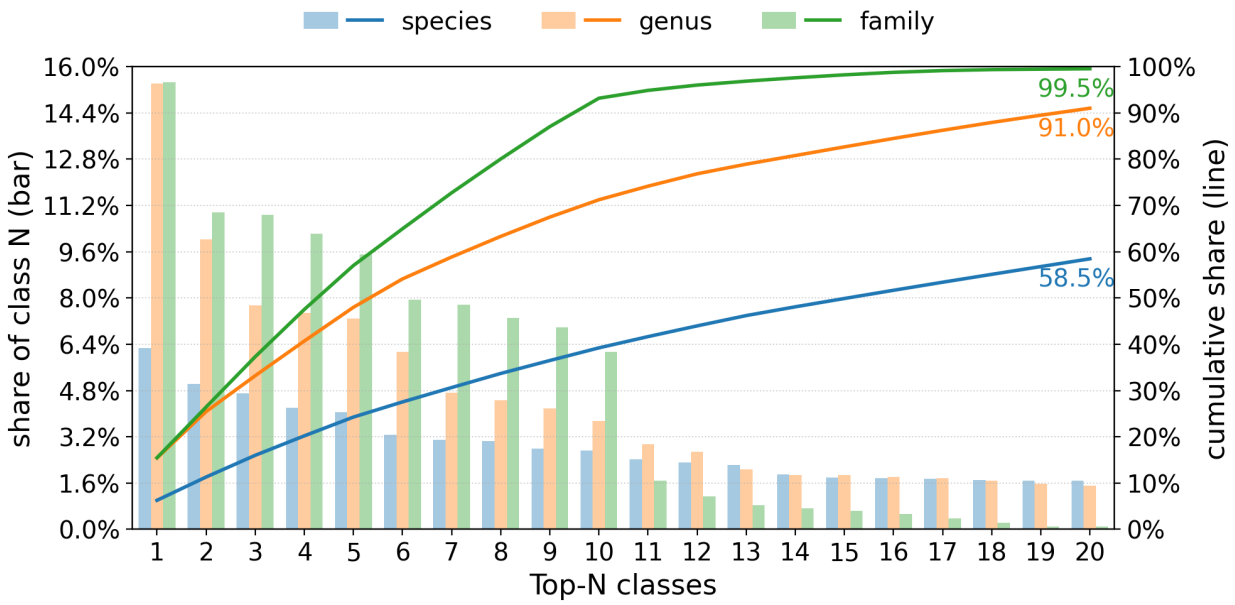


Figure 6.4: Top 20 class distribution across the multi-class taxonomy ranks. The long tail of rare classes is excluded from this plot. Bars show sample counts, and lines show cumulative class coverage. Aggregating labels from species to genus and family levels clearly increases balance, with most samples concentrated in a smaller number of classes. The binary leaf-type distribution is not shown here.

To avoid overfitting on extremely rare classes, only classes with at least 50 samples were kept for training. This threshold was chosen pragmatically, as classes with fewer than 50 samples were considered too sparse for a classifier to learn consistent patterns reliably. After removing these rare classes, the remaining set was further trimmed to include only the head classes covering 95% of the samples.

Table 6.2: Label aggregation and class coverage of the training data used (275,588 samples). Class coverage columns report the number of classes needed to cover the indicated percentage of the *total* dataset. Head/tail columns use the *labelled* denominator per rank. Rare classes are defined as classes that have < 50 samples.

	Sample count		**Class count (vs total)**				**Head/Tail class contribution**		
Label	**N**	**(%)**	**N**	**(95%)**	**(90%)**	**rare**	**top1 (%)**	**top5 (%)**	**rare (%)**
Species	239409	91.68	472	472	128	325	6.33	24.48	1.26
Genus	257064	98.44	126	29	21	68	15.55	48.38	0.25
Family	256431	98.19	51	13	10	23	15.60	57.49	0.08
Leaf_Type	256431	98.19	2	1	1	0	99.46	100	0.00

Table 6.2 summarises how aggregation changed class balance. The *rare (%)* column quantifies the long tail omitted from Figure 6.4, while the *N (95%)* and *N (90%)* columns show how many classes are needed to cover most of the training data. At the family level, for example, 13 classes cover 95% of the samples; the remaining 38 families were merged into a single *other* class to simplify the model while still retaining the full dataset coverage.

6.3. Feature Extraction and Model Setup

The features used to describe each segmented tree instance follow the set proposed by Chi et al. (2025) and are listed in Appendix C. Their study achieved high species classification accuracies when combining these geometric lidar features with hyperspectral data, which motivated their adoption here. However, in this study, the spectral component could not be used: the RGB values available in the AHN point cloud are draped from orthophotos, and were found to be misaligned with the 3D geometry (see Section 4.2.3). As a result, only the geometric and intensity-based descriptors from the Chi feature set were retained.

Due to the vegetation filtering procedure (Section 5.2), all first and last returns were removed, rendering the features `Ifirst_mean`, `Ilast_mean`, `Nfirst`, and `Nlast` unusable (NaN or zero for all samples). These four descriptors were therefore excluded from further analysis.

Pairwise Spearman rank correlations (Figure 6.5) show strong within-group correlations, especially in the height and intensity groups. A pilot ablation removing features with $|\rho| \geq 0.95$ before training caused a sharp drop in validation performance, suggesting that even highly correlated variables contribute complementary signals in combination with other descriptors. Given that Random Forests are comparatively robust to correlated predictors through bagging and random feature subspace selection, no aggressive feature pruning was applied. Model complexity was instead controlled through hyperparameter tuning (see Section 6.4).

Random Forest was selected as the classification algorithm because it is widely used in related work on vegetation classification, is relatively fast to train, and is known to handle correlated features well (Chehreh et al., 2023). It can combine diverse input descriptors without strong assumptions about their distributions and capture non-linear relationships between features and class labels, while remaining comparatively easy to train and interpret. Convolutional Neural Networks (CNNs) have achieved higher accuracy in similar studies (Hell et al., 2022; Chen et al., 2021a), but implementing such models was not feasible within the available time frame. The focus here was therefore not on achieving state-of-the-art accuracy, but on assessing whether taxonomy-based classification is feasible with the given dataset and feature set. Hyperparameter tuning was conducted to confirm optimal model performance; these settings are listed in Appendix E.

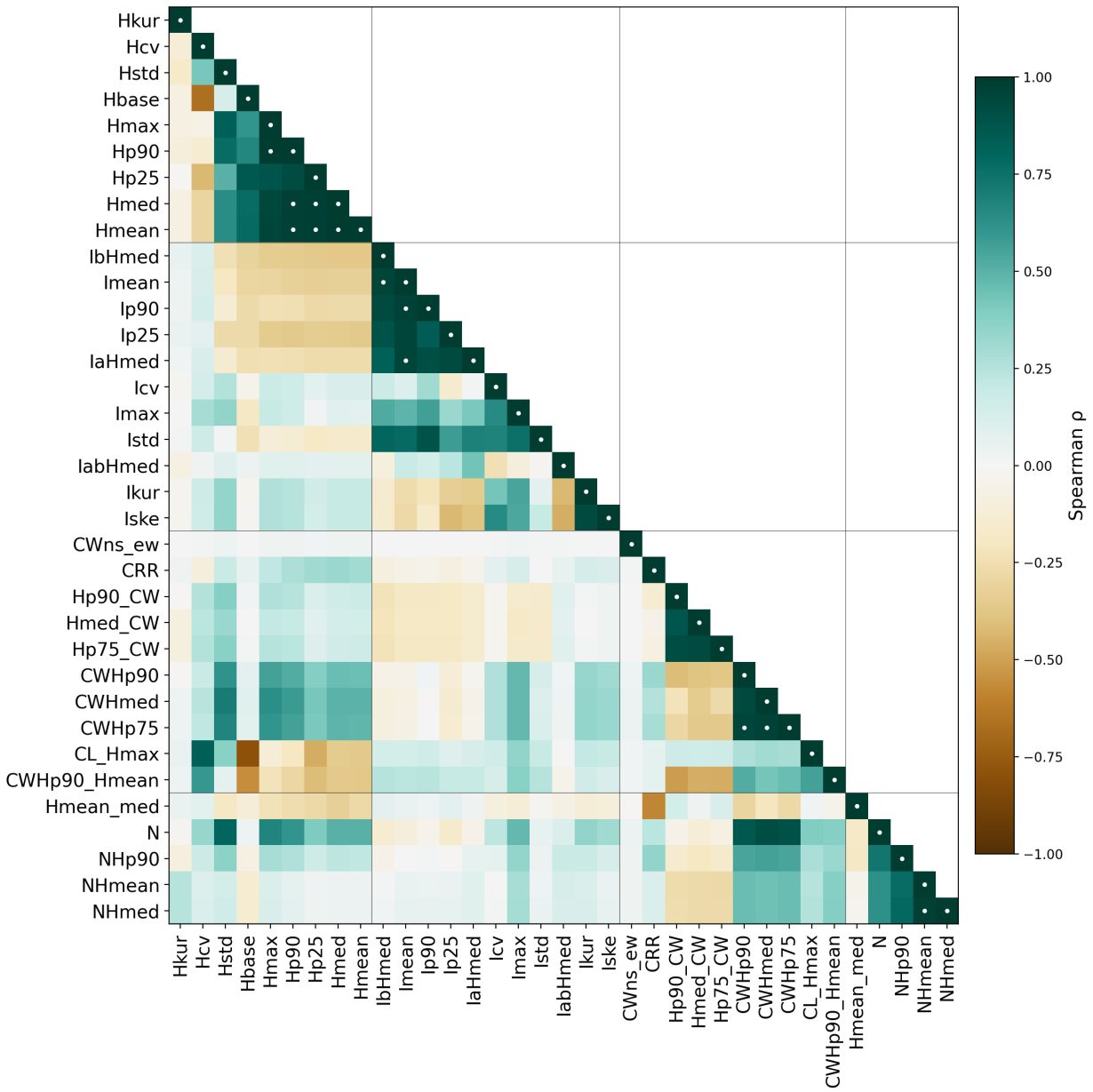


Figure 6.5: Spearman correlation matrix (lower triangle shown) for all retained features in the leaf_type classification task. Features are grouped by family (height, intensity, crown/shape, density). White dots mark feature pairs with $|\rho| \geq 0.95$, indicating strong within-group correlations. Despite this redundancy, no feature pruning was applied, as correlated variables were found to contribute complementary information during training.

6.4. Results and Performance Evaluation

For each taxonomic rank (species, genus, family, and leaf_type), a Random Forest model was trained and tuned. Model selection was based primarily on validation macro-F1, with balanced accuracy used to break ties and minority-class recall considered additionally for the binary leaf_type task.

Classification performance is reported using macro-F1, balanced accuracy, macro-recall, and micro-F1 (accuracy). Macro-averaged metrics treat all classes equally regardless of their size, making them more robust to class imbalance than micro-averaged accuracy, which weights larger classes more heavily. Balanced accuracy, defined as the mean of class-wise recall scores, further reduces the influence of dominant classes. All metrics are computed on held-out test sets and summarised in Table 6.3.

Table 6.3: Best Random Forest models and dataset statistics per taxonomy rank. Metrics are computed on the test set; selection by validation macro-F1 (ties by balanced accuracy and, for leaf_type only, minority-class recall).

Metric / Stat	Species	Genus	Family	Leaf Type
Macro F1	0.270	0.372	0.442	0.722
Balanced Accuracy	0.289	0.397	0.442	0.681
Macro Recall	0.289	0.397	0.442	0.681
Micro-F1 (= Accuracy)	0.356	0.439	0.496	0.995
CV Macro F1 (mean)	0.259	0.358	0.435	0.724
Minority Recall (binary)	-	-	-	0.363
ROC-AUC (binary)	-	-	-	0.054
Samples	238 552	258 628	258 438	258 640
Unique Classes	79	25	12	2
Class Sample Coverage	95%	95%	95%	100%
Train/Test split	80/20	80/20	80/20	80/20

Performance was generally low for all multi-class tasks, with macro-F1 scores below 0.45 even at the family level. While accuracy increased slightly at coarser taxonomic levels (from 0.36 at species to 0.50 at family), this is largely driven by the shrinking number of classes and does not reflect stronger discriminative power. The leaf_type classifier achieved a very high overall accuracy (0.995), but this is misleading due to the extreme class imbalance ($\approx$ 99.5% deciduous). Its balanced accuracy (0.68), macro-F1 (0.72), and minority-class recall (0.36) show that the model struggles to detect the rare evergreen trees reliably.

Figures 6.6 and 6.7 show the confusion matrices for the leaf type and family classifiers. The leaf type matrix highlights the skewed predictions toward the deciduous class, while the family matrix shows widespread confusion across classes, with only a few families being correctly identified at meaningful rates. Confusion matrices for all four classifiers (including species and genus) are provided in Appendix E.

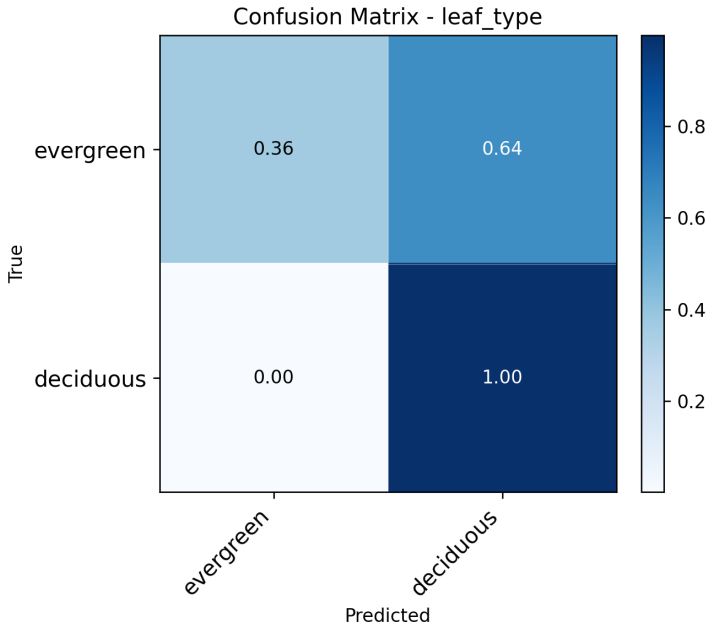


Figure 6.6: Confusion matrix of the leaf type classifier. Nearly all trees are predicted as *deciduous*, illustrating the effect of the extreme class imbalance: high overall accuracy but poor recall for the minority evergreen class (0.36). An enlarged version is available in Appendix E.

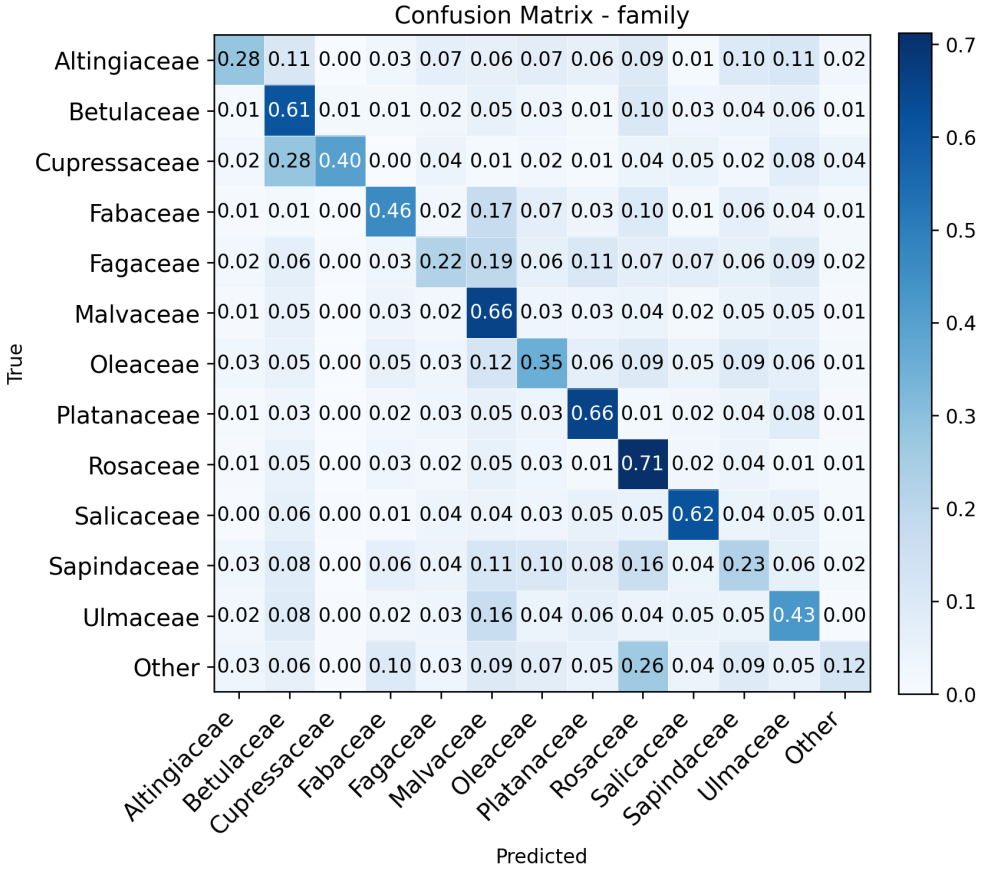


Figure 6.7: Confusion matrix of the family classifier. Predictions are scattered across many families, with only a few (e.g. Betulaceae, Platanaceae, Rosaceae) showing noticeable diagonal clusters, indicating limited class separability. An enlarged version is available in Appendix E.

6.5. Interpretation and Discussion

In summary, these results show that taxonomic classification from AHN point clouds using geometric and intensity features alone was not successful. Even at the coarser family level, the model fails to reach accuracy levels that would make it suitable for integration into the reconstruction pipeline.

Several factors contributed to this limited performance. First, the available training data were noisy and imbalanced. Labels were sourced from municipal inventories rather than field surveys, introducing errors (e.g. outdated names, placeholder entries, cultivar inconsistencies). As discussed in Section 2.3.3, this trade-off between label quantity and reliability is typical of inventory-based datasets, which provide broad coverage at the cost of taxonomic precision. To mitigate these inconsistencies, the inventory species names were automatically harmonised into a structured taxonomic hierarchy, allowing each sample to be represented at multiple ranks (species, genus, family). This aggregation enriched the label data and ensured that coarser classification tasks were derived systematically rather than manually defined.

Segmentation errors such as over- or under-segmentation could also have assigned incorrect labels, though this is not expected to be a dominant factor. More fundamentally, the dataset was heavily skewed, especially for the `leaf type` task where evergreens represented only 0.5% of samples. In addition, the coarsest multi-class model (family) still contained 12–13 classes plus an *other* group, whereas related studies often target only 3–6 classes. Attempting to separate twice as many classes using fewer feature types likely contributed to the low performance observed.

Beyond the limitations of the training data, a second factor concerned the feature set, which lacked colour information. Other studies have shown that tree species can be classified from point clouds when enough training samples are available and when colour information (e.g. NDVI or colour indices) is included alongside geometric descriptors. Chi et al. (2025), for example, reported high classification accuracies using a similar set of geometric features *combined* with hyperspectral data, whereas this study lacked reliable colour information. Here, colour data were excluded because the RGB values obtained from the orthophoto drape could not be trusted due to geometric misalignment between the 2D imagery and 3D point cloud. This left only geometric and intensity-based features, which appear insufficient to separate taxa reliably, especially when combined with the strong class imbalance in the data.

Third, the urban context introduces substantial intra-class variability. Trees of the same species can differ strongly in appearance depending on age, pruning practices, and planting conditions. Unlike forest environments, where most trees are mature and uniform, urban datasets capture a wide range of growth stages and structural forms. Additionally, very small trees were excluded by the vegetation filtering. While this was acceptable for the CFD use case, since such trees can be assumed to have negligible aerodynamic effects, it further skewed the dataset toward large and morphologically diverse samples.

Segmentation-derived label noise may have contributed to performance degradation, but it is not expected to be a dominant factor. The Random Forest algorithm itself was also not a limiting factor: hyperparameter tuning confirmed that the models operated near their optimal capacity, and the method is widely recommended in related work.

Based on these findings, taxonomy-based classification was not integrated into the reconstruction pipeline. With this dataset and feature set, the models cannot be trusted to provide reliable labels. Instead, porosity is now estimated empirically for each individual tree. This approach avoids errors from misclassification and may even better reflect reality, as trees of the same species can exhibit notable variation in crown structure and porosity.

6.6. Answer to RQ2 and Future Work

The preceding analyses and results provide a direct answer to the second research question:

> *Can urban trees be taxonomically classified from airborne lidar?*

The results indicate that taxonomy-based classification of urban trees from AHN point clouds is not feasible with the current dataset and feature set. This does not imply that taxonomy-based classification is inherently infeasible, but rather that it requires better data, labels, and features than were available within the time constraints of this thesis. Notably, the upcoming AHN6 release has announced the inclusion of RGB values derived from simultaneously acquired orthophotos, which would eliminate the current temporal mismatch between imagery and point cloud data—although geometric projection errors from mapping 2D images onto 3D points can still be expected.

Future work could address these limitations in several ways. Manually labelled and quality-controlled training datasets would reduce label noise and help balance the classes—for example, by deliberately collecting sufficient labelled evergreen samples to mitigate the current `leaf_type` imbalance. Including colour data—ideally acquired simultaneously with lidar and stored as native per-point attributes—would likely add crucial discriminative power. If such data are unavailable, stratifying the training data by age or size groups may reduce within-class variability and allow models to learn more consistent patterns. Preliminary observations suggested that some species occupy distinct height ranges (e.g. *Platanus* × *hispanica* occurring mainly among tall trees), so incorporating such contextual variables or age attributes where available could help reduce confounding effects. Finally, exploring deep learning methods such as convolutional neural networks could be worthwhile, as these models have achieved higher accuracies than Random Forests in similar studies using manually labelled datasets (Hell et al., 2022; Chen et al., 2021a).

7. TREE GEOMETRY RECONSTRUCTION FOR CFD SIMULATIONS

7.1. Introduction and Motivation

Urban trees influence the urban microclimate through aerodynamic drag and evapotranspiration, yet are rarely represented explicitly in city-scale digital twins. To enable their inclusion in computational fluid dynamics (CFD) simulations, this study reconstructs individual tree models from segmented ALS point clouds. Each model combines a porous crown and a solid trunk, reflecting the drag-imposing behaviour of these components. The reconstruction strategy prioritises geometric realism sufficient to influence airflow while remaining computationally efficient at city scale. The following sections describe the reconstruction of crown geometry using alpha wrapping, the estimation of volumetric porosity from point cloud structure, the derivation of trunk geometries from allometric relations, and the integration of these elements into CFD-ready CityJSON tree objects.

7.2. Crown Geometry Reconstruction Method

7.2.1. Crown envelope via 3D Alpha Wrapping

For CFD and volumetric analyses, a watertight crown envelope is required so that interior queries (e.g., void–to–point distances in Section 7.3) are well defined. Each crown is reconstructed with CGAL's 3D Alpha Wrapping (AW3) (Alliez et al., 2025), which produces a closed triangular surface and—unlike classical alpha shapes—guarantees a manifold, watertight envelope (see Section 2.4.2 for a brief contrast).

Before wrapping, the raw point cloud is vegetation-filtered and instance-segmented (see Section 5); the wrap is computed on the filtered canopy points of each tree. This suppresses outliers and makes the surface approximate the dense crown core rather than stray lidar returns. Tree point-cloud instances containing fewer than 50 points are excluded from reconstruction, as alpha wrapping on such sparse inputs can produce degenerate or unstable crown geometries. Trees removed at this stage are not passed to subsequent steps such as porosity or trunk estimation and are therefore absent from the final CityJSON output.

AW3 is controlled by two parameters: α (feature scale) and δ (offset/tightness) (Alliez et al., 2025). Following the AW3 manual, the offset is set as a small fraction of α so that α is the main level-of-detail knob. Both parameters are normalised using the bounding-box diagonal D of the tree point cloud:

$$\alpha = \frac{D}{r_\alpha}, \qquad \delta = \frac{\alpha}{r_\delta} = \frac{D}{r_\alpha r_\delta}.$$

Keeping the ratio $\delta/\alpha = 1/r_\delta$ fixed yields consistent tightness across tree sizes and avoids regimes where an offset that is too small or too large inflates mesh complexity. Practically, increasing r_α decreases α (finer/more connected wraps), while increasing r_δ decreases δ (a tighter fit).

To determine suitable values for the two control parameters, a coarse parameter sweep was conducted across N=100 randomly selected crown instances. The sweep evaluates how combinations of r_α and r_δ influence wrap connectivity and mesh complexity, providing empirical guidance for selecting globally stable values. The resulting parameter grid is shown in Figure 7.1.

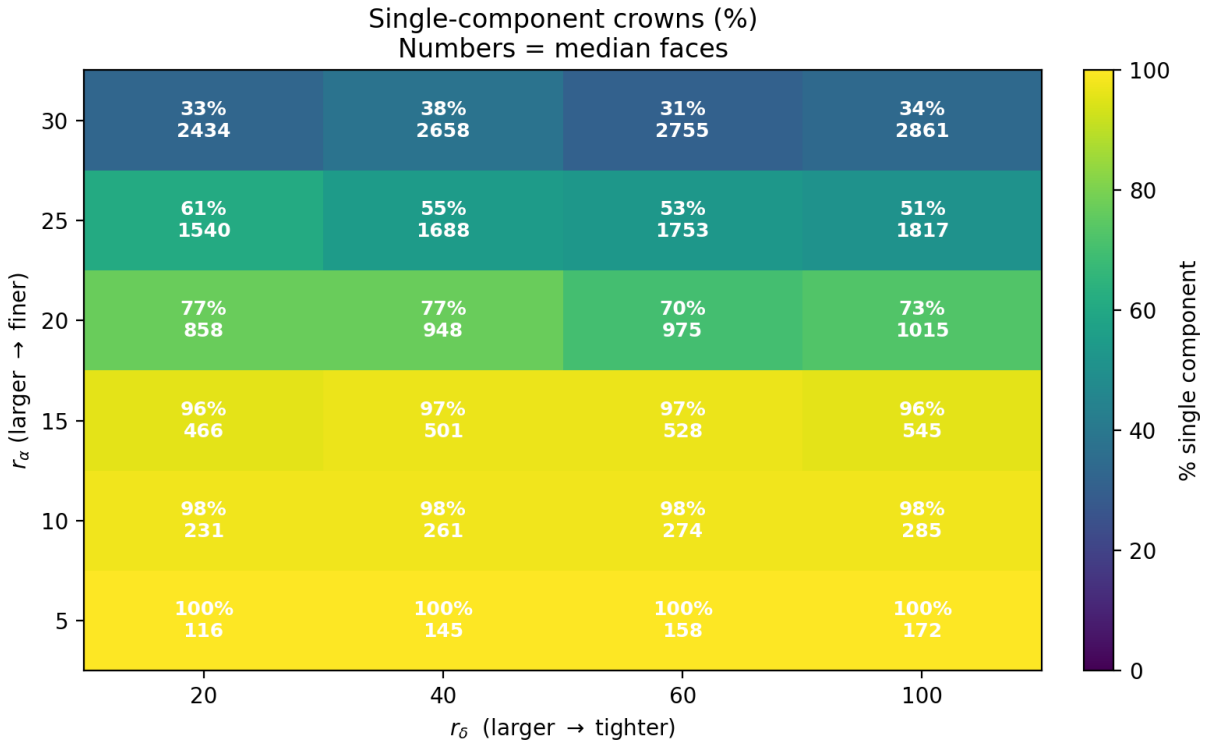


Figure 7.1: Coarse sweep of AW3 relative parameters over N=100 random crowns. Colours indicate the percentage of single-component wraps (i.e., a single connected triangle mesh); numbers show the median face count. Connectivity is driven by r_α; r_δ mainly fine-tunes tightness and mesh size. The chosen AW3 parameters are r_α=15, r_δ=50 (near-universal single component with lightweight meshes).

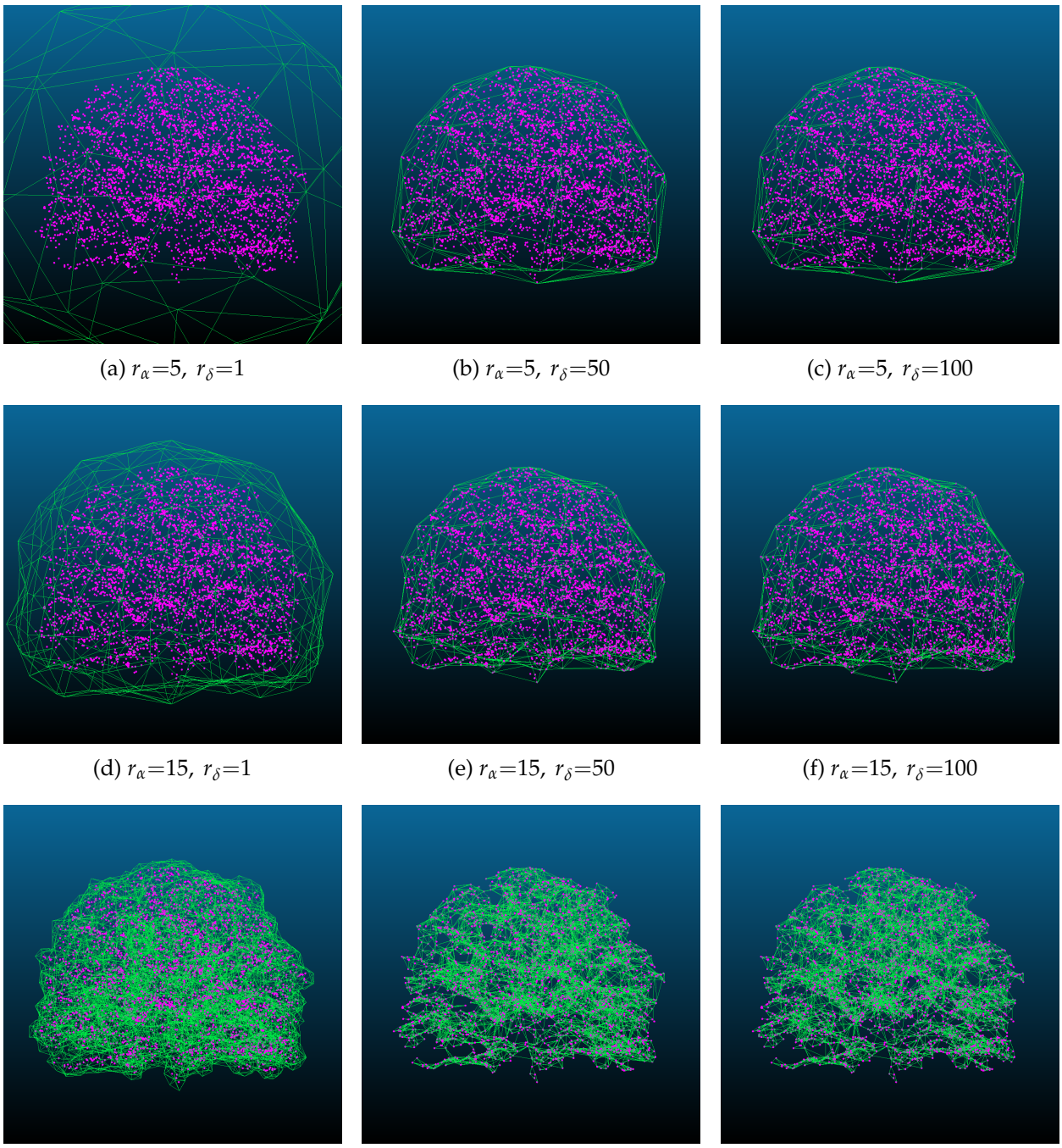
The results of the parameter sweep in Figure 7.1 indicate a clear balance point consistent with CFD meshing constraints: few components and low mesh complexity are preferred (see Section 2.1.5 for underlying modelling requirements). Values of $r_\alpha \leq 15$ yield $\gtrsim 97\%$ single-component wraps with only a few hundred faces on average, whereas larger values increase fragmentation and face count. The influence of r_δ is comparatively minor; within the tested range it mainly controlled how tightly the wrap followed the crown surface. Low r_δ values produced slightly inflated envelopes that overestimated crown volume, while high values fragmented the mesh or increased face count. A mid-range setting of r_δ=50 provided the most balanced outcome across all criteria (see also Figure 7.2), maintaining single-component connectivity, moderate mesh complexity, and a close yet stable fit around the canopy.

We therefore fix the AW3 parameters globally to:

$$r_\alpha = 15, \quad r_\delta = 50. \tag{2}$$

Occasional multi-component results are accepted, as the CFD objective favours robust, simple obstacles over highly intricate crowns.

Figure 7.2 complements the parameter sweep in Figure 7.1 by visualising how varying r_α and r_δ affects individual crown reconstructions. It qualitatively confirms the quantitative trend: finer wraps arise from lower r_α, while moderate r_δ values yield stable, well-fitted envelopes. The adopted setting (r_α=15, r_δ=50) lies near the optimal region identified in the sweep.



(a) r_α=5, r_δ=1 (b) r_α=5, r_δ=50 (c) r_α=5, r_δ=100

(d) r_α=15, r_δ=1 (e) r_α=15, r_δ=50 (f) r_α=15, r_δ=100

(g) r_α=50, r_δ=1 (h) r_α=50, r_δ=50 (i) r_α=50, r_δ=100

Figure 7.2: Effect of relative parameters on the *Alpha Wrapping* (AW3) envelope (green) created from the filtered tree point cloud (magenta). Rows increase r_α (larger $r_\alpha \Rightarrow$ smaller α, hence a finer/more connected wrap); columns increase r_δ (larger $r_\delta \Rightarrow$ smaller δ, i.e. a tighter fit). The centre panel (r_α=15, r_δ=50) is the setting adopted in this work; it sits near the knee in Fig. 7.1, balancing a high single-component rate with modest mesh complexity for CFD. This figure grid deliberately includes extreme settings (e.g., r_δ=1 and r_α=50) to illustrate behaviour at the limits; these extremes were not used in the parameter sweep of Fig. 7.1 nor in the final pipeline.

The resulting watertight envelope is used as the crown boundary for volumetric porosity estimation (Section 7.3) and for trunk placement (Section 7.4), ensuring an individualised, CFD-ready 3D model for each tree.

7.3. Crown Porosity Estimation

7.3.1. Volumetric Porosity Estimation using Adaptive Voxelisation

Volumetric crown porosity is typically estimated by discretising the crown volume into voxels and quantifying the fraction of empty cells within the crown envelope (Pfeiffer et al., 2018; Kamoske et al., 2019; Nowak et al., 2022). At a given voxel size h, the volumetric porosity is defined as

$$\varepsilon(h) = \frac{N_{\text{empty}}(h)}{N_{\text{total}}(h)}, \tag{3}$$

where N_{total} is the number of voxels inside the crown envelope and N_{empty} the number of those containing no canopy points.

However, such voxel-based estimates are highly sensitive to the chosen grid resolution: if voxels are too large, fine-scale gaps are merged and porosity collapses towards zero; if too small, the estimate reflects the discrete point sampling rather than the continuous tree structure. The influence of voxel size on porosity is illustrated in Figure 7.3. For illustration, the figure shows a 2D cross-section of the crown; the actual porosity estimation in the algorithm is performed in full 3D.

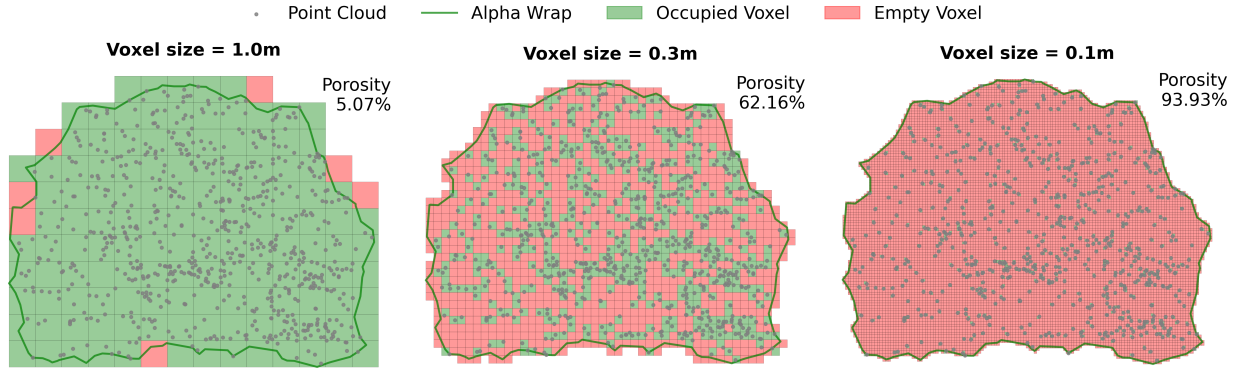


Figure 7.3: Effect of voxel size on crown porosity, shown on a 2D slice of the crown for visual clarity. The point cloud was downsampled to improve readability. With decreasing voxel size, the discretisation resolves smaller gaps, increasing the proportion of empty voxels. At extremely fine resolutions, porosity reflects point cloud sparsity rather than the tree's true structure. At large voxel sizes, most voxels become occupied, lowering porosity. In 3D, absolute porosity values are higher, but the same trends with voxel size are observed.

To obtain physically meaningful and reproducible porosity values, the voxel size should be linked to the tree's internal structure rather than chosen arbitrarily. Instead of prescribing a fixed resolution, the voxel size h is made adaptive by tying it to a characteristic length scale of the crown — the typical size of internal gaps between branches and foliage.

This structural scale is quantified using a sampling-based procedure inspired by Monte Carlo methods, which estimates the distribution of *empty–space distances* within the α–wrap envelope of the tree crown. To reduce stochastic variability, sample points were drawn *uniformly* within the envelope. For each sample, the Euclidean distance to the nearest canopy point was computed using a KD–tree search. The median of these distances, denoted d_{50}, represents the *characteristic gap size* of the crown — a single descriptive measure of the typical void radius within the canopy volume. The estimate converged within 0.1% for over 90% of trees at sample size, $N = 600{,}000$.

The resulting procedure and its geometric interpretation are shown in Figure 7.4, where uniformly distributed samples (blue) and their nearest–neighbour distances (red) visualise the concept of local void scale.

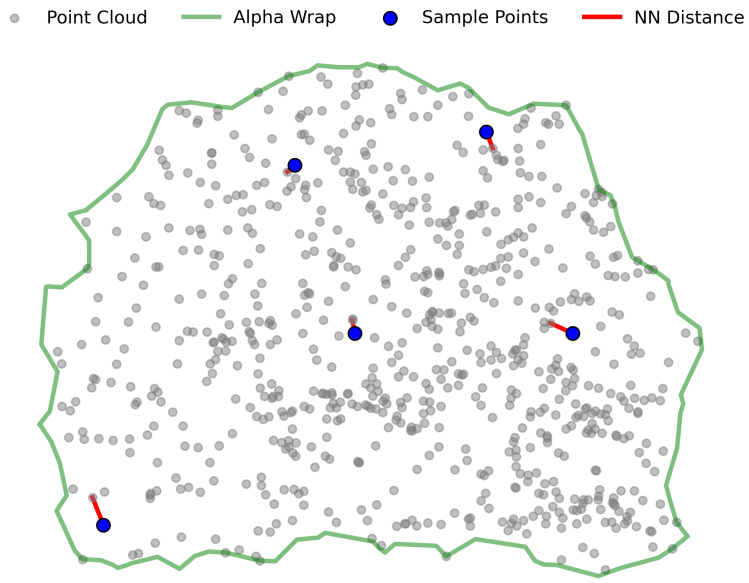


Figure 7.4: Illustration of the uniform sampling procedure inspired by Monte Carlo methods within the crown envelope. A 2D cross-section is shown for clarity, with downsampled canopy points. Sample points (blue) are uniformly drawn within the α–wrap, and distances to the nearest canopy points (red lines) represent the local void scale.

Knowing d_{50} provides a natural resolution for voxelisation: the voxel grid should be fine enough to capture gaps of this characteristic size, yet not so fine that it begins to reflect point-sampling noise. For example, if $d_{50} \approx 0.5$ m, a voxel size of 0.5 m would blur smaller openings, whereas 0.05 m would oversample the structure, inflate the porosity value, and greatly increase computational cost. The voxel size then is defined as a fraction of this gap scale:

$$h = k \cdot d_{50}, \qquad k \in (0, 1]. \tag{4}$$

This definition provides a principled balance: it selects the largest voxel size that still resolves the structural gaps representative of the crown. The scale factor k controls the voxel size relative to the crown's characteristic gap scale and is calibrated empirically in Section 7.3.2 to yield stable and physically consistent porosity estimates.

To avoid degenerate behaviour at extremely high or low resolutions, h is clamped between 0.02 m and 1.0 m. The lower bound safely captures the smallest structural gaps observed in AHN trees, including fine leaf-scale openings, while the upper bound acts as a coding failsafe and was never reached in practice.

Because the reconstructed crown geometry will be used in CFD as a single porous obstacle, the resulting porosity must represent an intrinsic, orientation-independent property of the tree. Previous fixed-resolution approaches, such as Pfeiffer et al. (2018), have shown that voxel-based porosity estimates can vary substantially with grid orientation, yielding different values for azimuthal, frontal, and lateral directions. Such dependency is undesirable, as it ties porosity to the discretisation rather than the tree's physical structure. In the following subsection, the scale factor k is calibrated by evaluating porosity across multiple grid orientations and offsets, and selecting the k at which orientation effects become negligible.

7.3.2. Calibration of the scale parameter k

To ensure that the porosity estimate is independent of grid orientation and phase, the scale factor k was calibrated on a separate set of 100 trees. For each tree and each candidate $k \in [0.5, 1.0]$, the crown was voxelised at $h = k \cdot d_{50}$ and porosity was computed across R=8 random 3D rotations and J=4 random grid offsets. The offsets were drawn uniformly within the voxel size range $[0, h)$ along each axis, meaning the grid origin was shifted by a random amount between 0 and h in x, y, and z directions. This produces 32 porosity values per tree and per k (one for each rotation–offset combination).

These 32 realisations do not themselves make the estimate rotation-invariant; they are used to measure how sensitive the porosity estimate is to grid orientation at each scale. Values below k=0.5 were excluded from the sweep, as such small voxel sizes tend to capture point-sampling noise rather than structural gaps, yielding artificially high porosity values approaching unity. If k is too low (small voxels), the estimate becomes noisy: slight grid shifts alter which voxels contain canopy points, causing artificial fluctuations. If k is too high (large voxels), most voxels are occupied and porosity collapses towards zero, losing structural meaning. The goal is to identify a k where porosity values are both plausible and stable across all orientations.

Two summary metrics were computed across the 100 trees for each k:

- **Median porosity** $M(k)$ (blue, with per-tree boxplots), representing the typical porosity magnitude across trees.
- **Median** 3σ **(99.7%) confidence interval width** $C(k)$ (orange, right axis), representing orientation sensitivity, computed as

$$C(k) = \text{median}\left(3\,\sigma / \sqrt{32}\right) \tag{5}$$

  where σ is the per-tree standard deviation across the 32 orientation/offset realisations. Dividing by $\sqrt{32}$ converts the orientation-wise standard deviation into the standard error of the mean, yielding a confidence interval for the mean porosity.

The optimal scale factor k was determined by balancing stability against physical plausibility according to two criteria:

1. **Stability:** $C(k) \leq 0.005$, indicating that orientation effects are negligible and porosity can be treated as an intrinsic property of the crown.
2. **Plausible magnitude:** $M(k) \in [0.85, 0.95]$, consistent with volumetric crown–scale porosities reported for trees in CFD studies (see Section 2.1.3), and lying above typical 2D optical porosities of windbreaks ($\sim$0.2–0.4) but below fine–scale void fractions ($\sim$0.95–0.99).

The two criteria above were evaluated across all candidate k values in a calibration sweep over the 100-tree dataset. Figure 7.5 summarises the results, showing how the median porosity $M(k)$ and its orientation sensitivity $C(k)$ vary with scale.

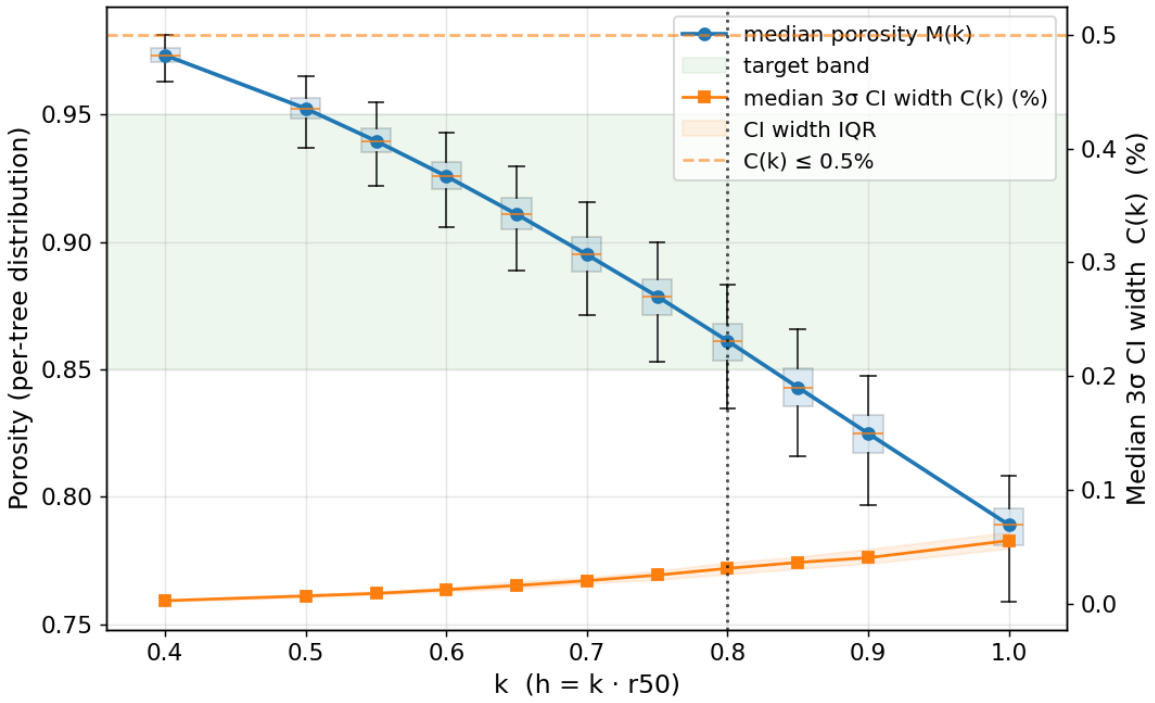


Figure 7.5: Calibration sweep over k on N=100 trees. **Blue:** per-tree porosity distributions (boxplots) with median $M(k)$ connected. **Orange (right axis):** median 3σ confidence interval width $C(k)$ across 32 grid orientations. The green band marks the CFD-consistent porosity target $[0.85, 0.95]$; the orange dashed line marks the stability criterion $C(k) \leq 0.5\%$. The vertical dotted line highlights the selected k=0.80. *An additional case at k=0.4 is included to illustrate the limiting behaviour for excessively fine voxel sizes; this value lies outside the calibrated range.*

As illustrated in Figure 7.5, orientation sensitivity remains consistently low across all tested scales: the orange line ($C(k)$) stays far below the stability threshold of 0.5% for all k, indicating that the porosity estimates are effectively rotation-invariant. Meanwhile, the blue boxplots show a clear decline in porosity with increasing k, as expected when voxel size increases and more voxels become occupied. Among the tested values, k=0.80 provides the largest voxel size whose interquartile ranges remain fully within the target band derived from literature, making it the optimal balance between computational efficiency (fewer voxels) and accuracy (physically plausible porosity values).

This calibration establishes $h = 0.8 \cdot d_{50}$ as a robust scale for capturing the structural porosity of the crown while eliminating orientation dependence. The resulting rotation-invariant porosity scalar forms a key canopy descriptor for CFD modelling and complements the geometric reconstruction of the crown.

7.4. Trunk Geometry Reconstruction Method

The trunk is represented as a simple cylindrical solid that provides a geometric connection between the crown and the ground surface (see Sections 2.1.2 and 2.1.5 for the CFD distinction between solid trunks and porous crowns). Although aerodynamic drag is primarily governed by the porous crown volume, the trunk influences near-ground flow patterns and ensures geometric continuity for meshing and CFD simulations. Because the lower stem is often occluded in ALS point clouds, its geometry cannot be measured directly and must be inferred from surrounding features.

Each trunk is therefore procedurally generated as a cylinder anchored to the Digital Terrain Model (DTM) at its base and oriented toward the centroid of the reconstructed crown. Its dimensions are derived from geometric and allometric relations based on the crown shape and local ground elevation. Following the principle demonstrated by Fu et al. (2020), who showed that crown dimensions and tree height are reliable predictors of stem diameter, this approach estimates trunk size using crown-based metrics rather than direct stem observations.

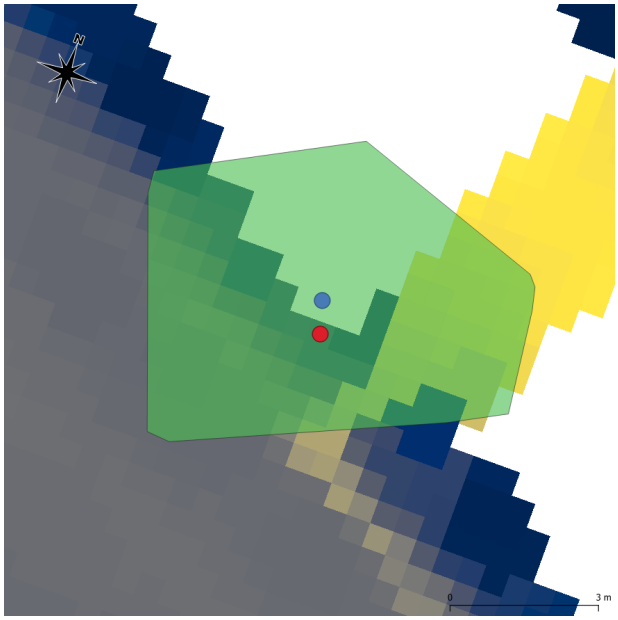
7.4.1. Base positioning using DTM

The coordinate of the trunk base is determined from the Digital Terrain Model (DTM) derived from the same ALS point clouds (0.5 m resolution). The algorithm starts from the crown centroid, defined as the mean horizontal position of the crown mesh (x_c, y_c) combined with the median crown height z_{med}. The objective is to find the nearest valid ground elevation from the DTM.

To this end, the crown mesh is projected onto the horizontal plane, and its convex hull defines the crown footprint polygon. The DTM is cropped to this footprint, and all valid elevation pixels within the area are collected. The DTM pixel whose horizontal coordinates (x_b, y_b) lie closest to the crown centroid is selected as the trunk base, with its elevation value assigned as z_{base}. If no valid DTM pixels exist within the footprint, the trunk is omitted from the tree model.

This procedure accommodates tilted or displaced crowns, allowing the base to deviate horizontally from the centroid while maintaining a realistic ground attachment. Figure 7.6 illustrates the concept: panel (a) shows a real-world example motivating the approach, and panel (b) visualises the DTM-based snapping procedure used to position the trunk base within the crown footprint.

(a) Google Street View image of a tilted street tree, where the trunk base is horizontally offset from the crown centroid.



(b) Top view of the DTM-based snapping procedure showing the crown convex hull projection (green) and selected trunk base (red).

Figure 7.6: Illustration of the trunk base positioning method. (a) shows a real-world example from Google Street View that motivated the approach. (b) visualizes how the nearest valid DTM pixel within the crown footprint is selected as the trunk base when no ground value is defined directly beneath the crown centroid; the blue point marks the crown centroid, and the red point indicates the selected trunk base.

7.4.2. Trunk dimensions and orientation.

The trunk dimensions are derived from the geometric relationship between the previously defined crown centroid (x_c, y_c, z_{med}) and trunk base (x_b, y_b, z_{base}). The trunk height is then defined as the vertical separation between the crown centroid and the trunk base, $H = z_{med} - z_{base}$, while the crown width is derived from the convex-hull area A, expressed as an equivalent circular diameter, $CW = 2\sqrt{A/\pi}$. The actual trunk length (L) is calculated as the three-dimensional Euclidean distance between the base and the crown centroid, ensuring realistic trunk geometry even when the crown is tilted or horizontally displaced. The trunk diameter at breast height (DBH) is then estimated through an allometric relation that links crown width and height:

$$\mathrm{DBH}_{cm} = a \cdot (\mathrm{CW})^b \cdot H^c, \tag{6}$$

where $a = 1.0$, $b = 1.1$, and $c = 0.7$ are empirically chosen parameters (Fu et al., 2020). Because DBH is conventionally expressed in centimetres in forestry literature, it is converted to metres when defining the trunk radius for geometric reconstruction: $r_{trunk} = \frac{1}{2} \cdot \mathrm{DBH}_{cm}/100$. Species-specific or slenderness-constrained formulations were not applied, as the objective was to obtain physically plausible but generalised dimensions suitable for aerodynamic modelling.

Figure 7.7 illustrates the geometric relationships between the trunk base, crown centroid, and derived parameters (H, CW, and DBH) used in the reconstruction process.

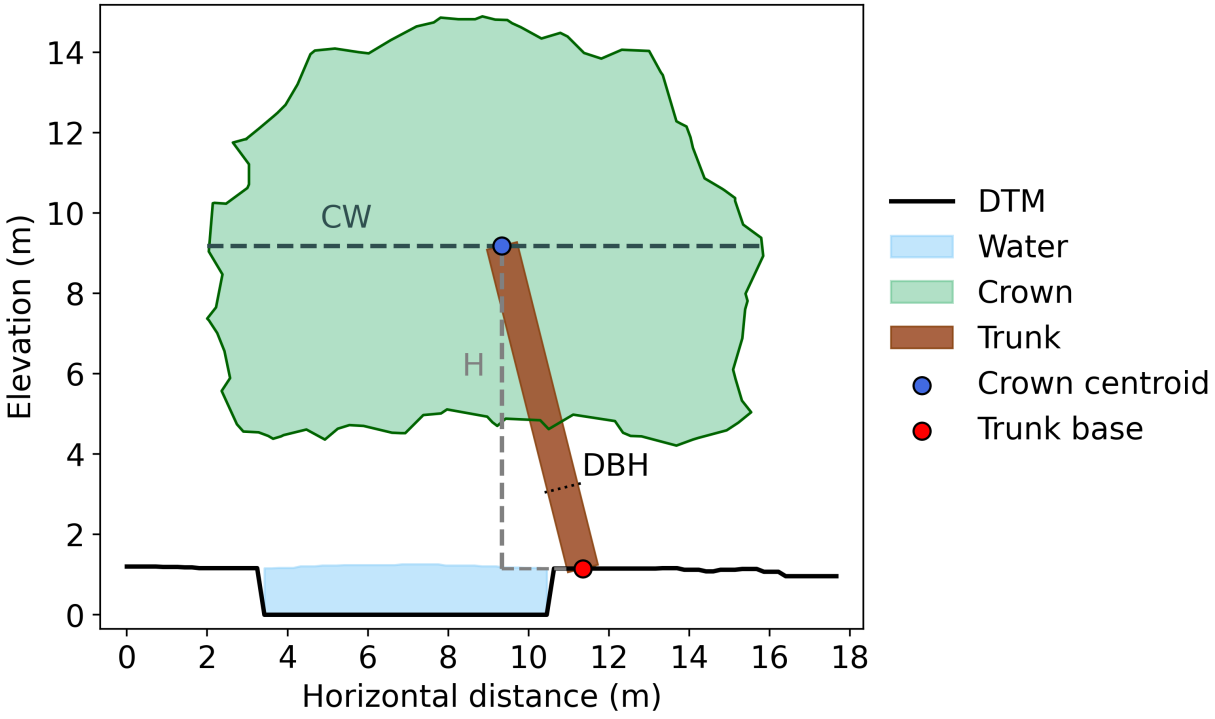


Figure 7.7: Schematic side view illustrating the geometric relationships defining trunk dimensions. The crown width (CW) is derived from the two-dimensional convex hull projection of the crown, while the apparent height (H) is defined as the vertical distance between the crown centroid (blue) and trunk base (red) coordinates. The crown, trunk, and digital terrain model (DTM) geometries are derived from a real segmented tree point cloud. A simplified water section was manually added to the DTM to exaggerate the apparent trunk tilt for clarity. Although the water is not present in the actual data, the resulting configuration preserves the true geometric proportions as calculated by the algorithm.

The trunk axis is represented by the three-dimensional vector connecting the base and crown centroid, along which a cylindrical geometry of radius r_{trunk} and length L is generated. The cylinder is discretised into 32 lateral faces to ensure a smooth visual appearance while maintaining manageable mesh complexity. When the crown centroid and base are horizontally offset, the cylinder is inclined accordingly, resulting in a tilted trunk representation.

7.5. Export of CFD-Ready Tree Models

All reconstructed tree instances are exported in `CityJSON` format, chosen for its lightweight structure and native support for semantic attributes. An example of the exported dataset and its attribute structure is shown in Figure 7.8. Each tree is stored as a `SolitaryVegetationObject` with a unique identifier (GTID) and an accompanying crown porosity attribute. CityJSON also allows compact storage of triangulated geometries (as used for the crown and trunk models), making it well-suited for large urban-scale datasets.

For CFD preparation, the export distinguishes between the two geometric components of each tree: the **crown**, exported as a porous volume mesh annotated with its volumetric porosity scalar, and the **trunk**, exported as a solid cylinder with zero porosity.

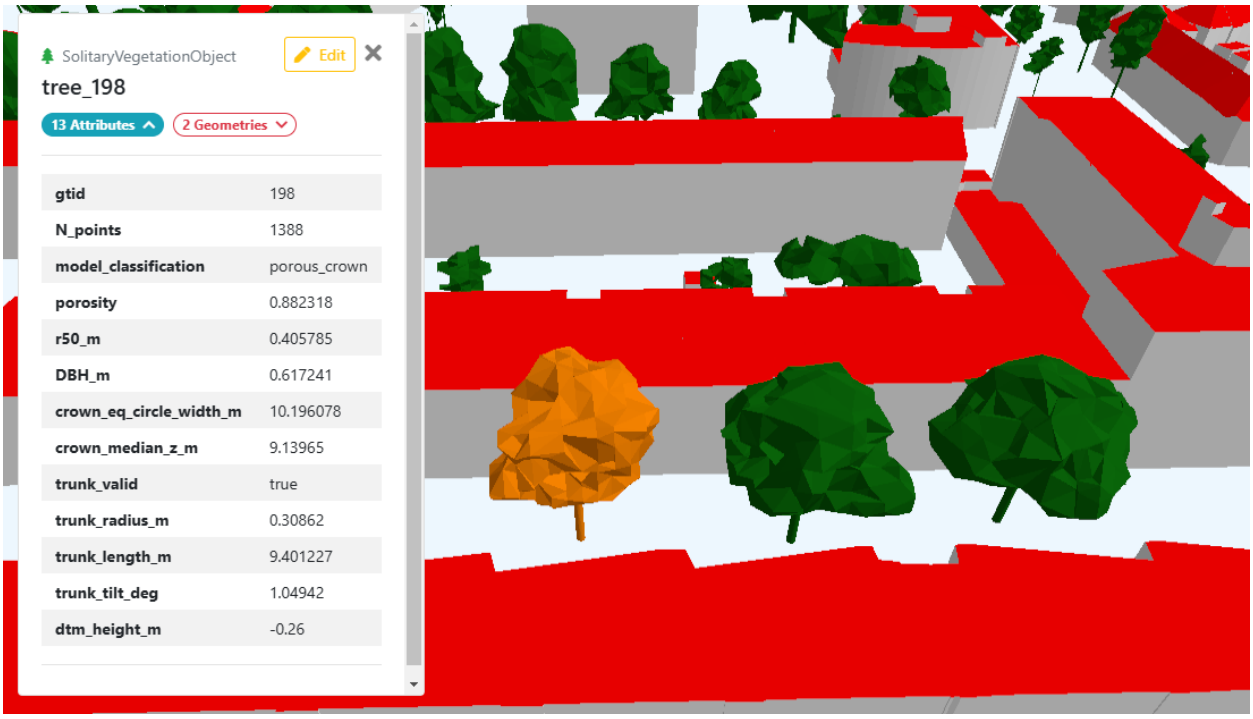


Figure 7.8: Visualisation of reconstructed tree instances exported in `CityJSON` format. Each tree is represented as a `SolitaryVegetationObject` containing semantic attributes such as the global tree identifier (GTID), crown porosity, and geometric parameters (e.g. crown width, trunk radius, trunk length, and tilt). The crown is modelled as a triangulated porous volume (alpha shape), while the trunk is represented as a solid cylinder. The selected tree (orange) and its attributes are displayed in a 3D viewer to illustrate the data structure and level of detail provided for computational fluid dynamics (CFD) applications. For spatial context, the surrounding **LoD 2.2** buildings are taken from the open **3DBAG** dataset by the *3D Geoinformation Research Group (TU Delft)* and *3DGI*, licensed under **CC BY 4.0**.

All trees are provided together in a single CityJSON file. CFD practitioners can subsequently group or split the dataset according to their modelling needs (e.g. by porosity bins or vegetation classes). This keeps the reconstruction pipeline agnostic to specific CFD workflows while still supplying all information required for aerodynamic modelling.

Although only porosity is used for CFD in this study, the CityJSON structure supports including additional tree-level attributes (e.g. estimated leaf area density, species label, or biomass) for future extensions. This ensures the exported models can serve as a flexible interface between the reconstruction pipeline and different environmental simulation frameworks.

7.6. Interpretation and Discussion

This section interprets the reconstructed geometries in terms of their realism, robustness, and practical suitability for city-scale CFD simulations.

7.6.1. Representation of canopy geometry

The use of alpha wrapping for canopy reconstruction yielded watertight crown models approaching LoD 3B realism while remaining computationally efficient. Each crown forms a smooth outer envelope that captures the dominant drag influence of the tree volume, in contrast to many digital twin implementations where vegetation is simplified or omitted. Particular attention was given to the requirements of CFD meshing: excessively complex or fragmented geometries tend to increase computational cost and complicate mesh generation around the trees. The alpha wrapping parameters were therefore selected to favour single-component crowns with a moderate number of triangular faces, providing a consistent balance between geometric realism and meshing robustness. Although this approach inevitably bridges fine structural gaps between branches and leaves, it ensures a geometrically coherent and watertight surface suitable for stable CFD simulation.

7.6.2. Porosity estimation and drag consistency

Drag behaviour is governed not only by crown shape but also by its porosity. Traditional voxel-based porosity estimates are highly sensitive to voxel resolution and grid alignment. In this study, a Monte Carlo approach was implemented where the optimal voxel size is expressed as a constant k times the characteristic gap size (d_{50}) of the tree point cloud. The selected value $k = 0.8$ produced rotationally stable results within the 0.85–0.95 porosity range commonly reported in the literature.

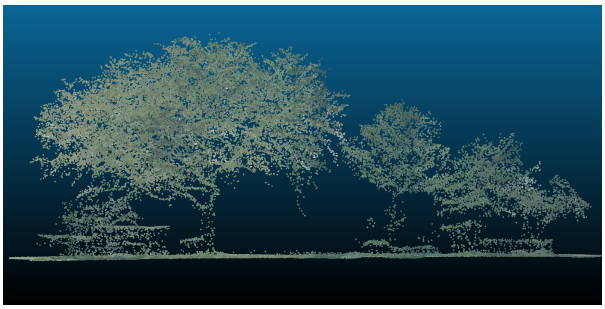
This formulation adapts automatically to point cloud resolution, since d_{50} reflects the inherent spacing between returns. Consequently, the resulting porosity is robust to variations in lidar acquisition density (e.g., ALS versus TLS). Each crown is treated as a homogeneous porous medium, a simplification that neglects intra-crown porosity gradients but ensures direction-independent drag coefficients appropriate for city-scale CFD.

Future work could extend this approach by partitioning crowns into multiple subregions with locally varying porosity values, enabling a more spatially detailed drag representation. One potential strategy is to deliberately oversegment trees into branch-level clusters, reconstructing each as an independent crown element. Such sub-crown representations would allow porosity estimation per structural component and open the possibility of refining trunk reconstruction as well — by extending the current single-cylinder model to include secondary branch cylinders that connect back toward the shared trunk base.

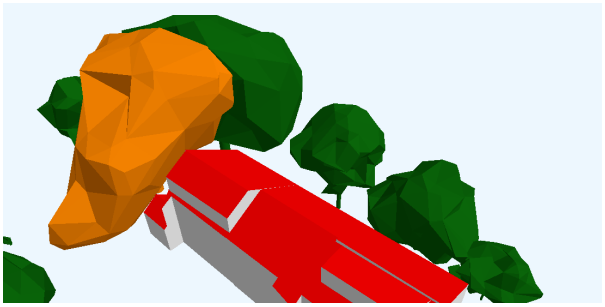
7.6.3. Propagation of segmentation errors

Upstream segmentation quality directly influences the reconstructed geometry and, consequently, the drag interpretation.

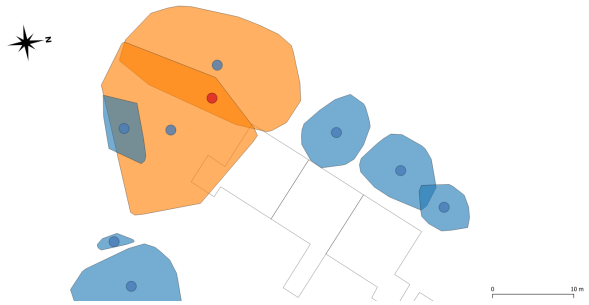
In the case of **oversegmentation**, a single tree may be divided into several smaller crown models (Figure 7.9). Each of these partial crowns produces its own thin trunk because the crown dimensions are underestimated relative to the true tree. Although this leads to multiple hallucinated trunks, the total crown volume across all segments still approximates the true canopy volume. From a CFD perspective, this is acceptable, as the flow obstruction represented by the total crown mass remains realistic. In fact, porosity estimates can even become more accurate, since alpha wrapping each smaller segment excludes inter-branch voids that would otherwise be included in a single large crown.



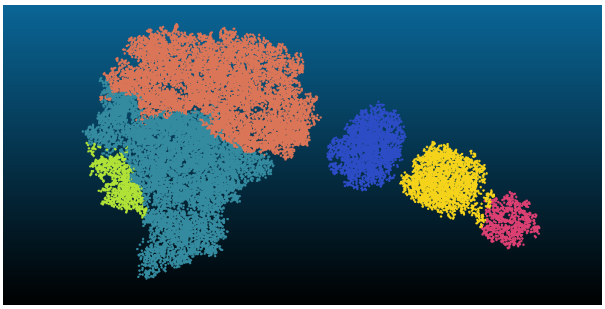
(a) Raw point cloud (side view)



(b) Reconstructed geometry with buildings



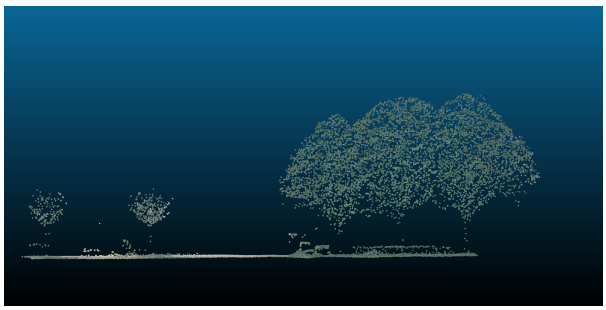
(c) Crown hull footprints (top view)



(d) Segmented point cloud clusters (top view)

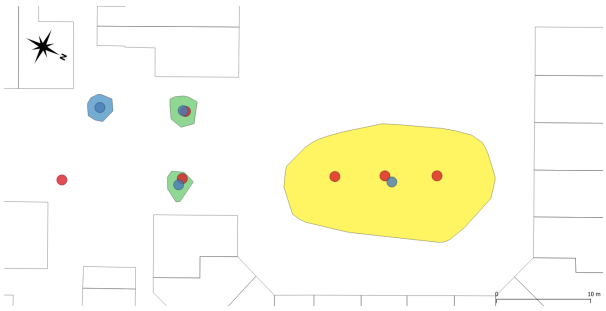
Figure 7.9: Example case of **oversegmentation** in the reconstruction pipeline. A single tree is divided into multiple smaller crowns, each generating a thin trunk due to reduced crown dimensions. The total canopy volume remains consistent with the original tree, preserving overall aerodynamic obstruction, but the trunk count is artificially increased. (a) Raw ALS point cloud colourised with RGB values from geotiles; (b) Reconstructed geometry with LoD 2.2 buildings from 3DBAG for context; (c) Crown hull projection showing centroid (blue) and municipality trunk location (red); (d) Segmented point cloud showing separate sub-crown clusters.

Conversely, **undersegmentation** merges multiple neighbouring trees into a single composite crown (Figure 7.10). This enlarges the reconstructed mesh and increases the calculated porosity, as the void space between the original trees becomes enclosed within the alpha wrap. A single, oversized trunk is then placed at the combined centroid, misrepresenting the individual stems. While geometrically inaccurate, the main flow obstructing volume—the total crown space—remains present in the model, albeit simplified as one broader tree.

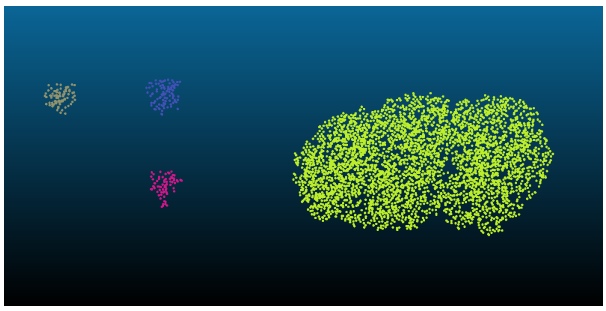


(a) Raw point cloud (side view)

(b) Reconstructed geometry with buildings



(c) Crown hull footprints (top view)

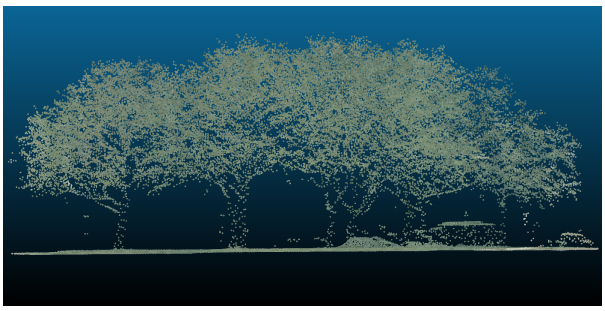


(d) Segmented point cloud clusters (top view)

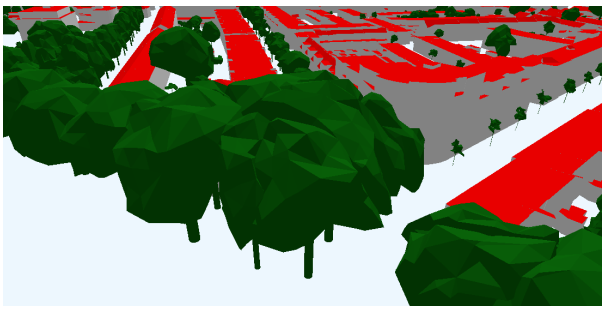
Figure 7.10: Example of **undersegmentation** in the reconstruction pipeline. Multiple trees are merged into a single crown, increasing apparent porosity and producing one oversized trunk at the merged centroid. (a) Raw ALS point cloud colourised with RGB values from geotiles; (b) Reconstructed geometry with LoD 2.2 buildings from 3DBAG for context; (c) Crown hull projection showing centroid (blue) and municipality trunk location (red); (d) Segmented point cloud showing the merged cluster.

Overall, both segmentation errors primarily affect trunk placement and porosity, while the representation of the crown volume—the dominant drag-inducing component—remains largely preserved. Mixed clusters, where both effects occur simultaneously, therefore still provide a plausible approximation of the overall canopy mass, though with less realistic trunk distributions and local porosity variations (Figure 7.11). These findings suggest that aerodynamic reliability is more resilient to segmentation inaccuracies than geometric fidelity, but that improved segmentation would nonetheless enhance physical realism in future iterations.

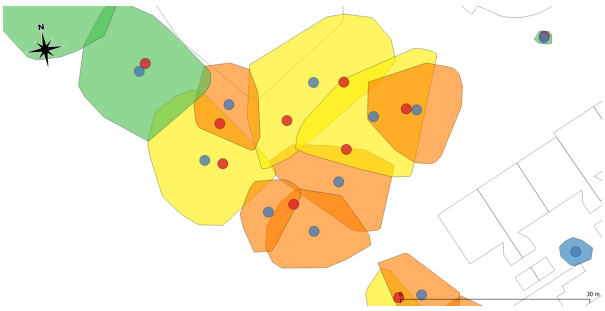
These segmentation-derived offsets are further reflected in the overall placement accuracy of reconstructed trees, as examined in the next subsection.



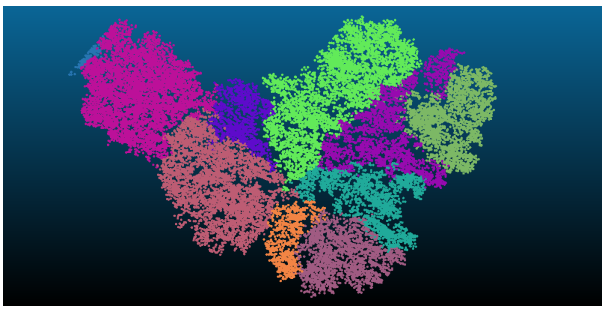
(a) Raw point cloud (side view)



(b) Reconstructed geometry with buildings



(c) Crown hull footprints (top view)



(d) Segmented point cloud clusters (top view)

Figure 7.11: Example case of a **mixed cluster** showing both oversegmentation (OS) and undersegmentation (US) effects in the reconstruction pipeline. Several neighbouring trees are partially merged while others are subdivided into smaller crowns, affecting crown porosity estimation as well as trunk count and size. Despite these geometric inconsistencies, the total crown volume remains well represented, preserving the aggregate aerodynamic obstruction. (a) Raw ALS point cloud colourised with RGB values from geotiles; (b) Reconstructed geometry with LoD 2.2 buildings from 3DBAG for context; (c) Crown hull projection showing multiple centroids (blue) and municipality trunk locations (red); (d) Segmented point cloud showing both merged and subdivided tree clusters.

7.6.4. Trunk reconstruction and placement accuracy

Trunks were reconstructed by connecting each crown to the digital terrain model (DTM) through a cylindrical geometry. The cylinder is defined between the crown centroid and the nearest DTM coordinate beneath it, ensuring that trunk position and height are derived consistently from the same ALS point cloud. This single-source approach prevents inconsistencies that could arise from combining external elevation data with vegetation geometry.

The trunk diameter at breast height (DBH) was estimated from crown height and width using a simplified empirical relationship inspired by Fu et al. (2020). While Fu et al. proposed a mixed-species allometric model for forest environments, the present study applies a reduced formulation that captures the proportional link between crown dimensions and trunk thickness without relying on species-specific coefficients. This choice was motivated by the limited reliability of the taxonomic classification results discussed in Section 6, which prevented species-dependent modelling. The simplified formulation preserves scalability and computational efficiency while maintaining physically plausible proportions for urban trees of varying sizes.

Each trunk is represented as a straight vertical cylinder extending from the DTM surface to the crown centroid. This simplified geometry does not account for natural curvature or branching but provides a stable aerodynamic obstruction for CFD simulations. Intersections between the trunk

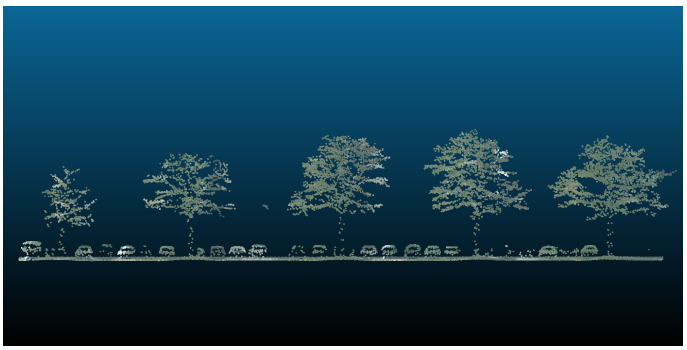
and crown alpha mesh occur because the two components are reconstructed independently; while these overlaps may locally affect meshing quality, they were not resolved further due to time constraints.

The trunk base is currently defined using the DTM coordinate closest to the crown centroid, which provides a reasonable geometric estimate but not necessarily the true stem position. As illustrated in Figures 7.9, 7.10, 7.11 and 7.12 (blue versus red markers in subfigure c), this approach can deviate from the trunk locations recorded in municipal tree inventories. Future work could improve realism by linking reconstructed crowns to the corresponding municipal trunk coordinates and aligning the trunk base with the DTM height at that location. While feasible for isolated public trees, this would be more challenging in oversegmented or undersegmented clusters where crown-to-record matching is ambiguous, and impossible for unregistered private trees absent from the database.

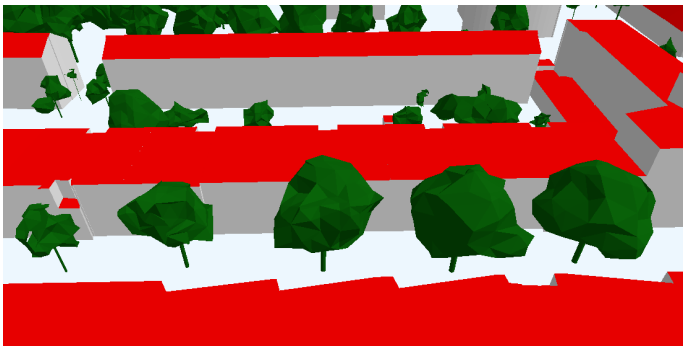
Overall, the current method yields consistent and physically reasonable trunk placements relative to the reconstructed crowns, suitable for wind modelling at city scale.

7.6.5. Overall reconstruction implications

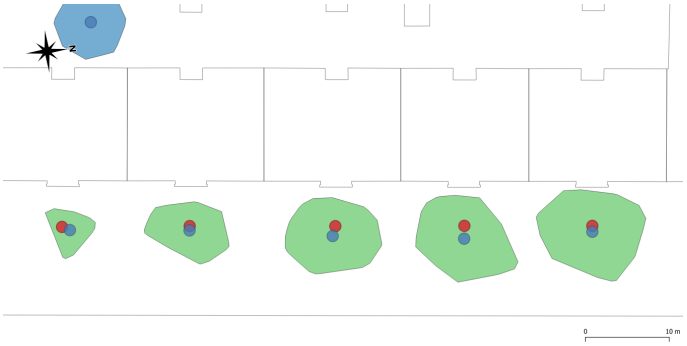
The reconstructed geometries, though simplified, successfully capture the dominant drag behaviour of urban trees. When segmentation performs well, the resulting crown and trunk models align closely with the true canopy structure and produce realistic representations suitable as drag-imposing geometries in urban CFD (Figure 7.12). The watertight crowns ensure reliable meshing, the porosity model provides physically interpretable drag scaling, and the trunk representations capture near-ground blockage effects essential for urban flow modelling. Taken together, these design choices prioritise numerical stability and consistency over botanical detail, yielding a robust and scalable framework for representing vegetation in city-scale CFD simulations.



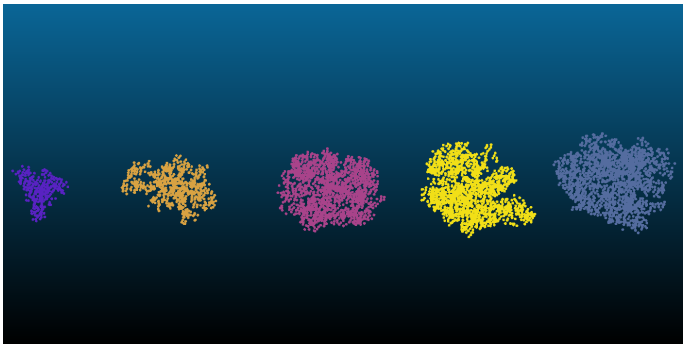
(a) Raw point cloud (side view)



(b) Reconstructed geometry with buildings



(c) Crown hull footprints (top view)



(d) Segmented point cloud clusters (top view)

Figure 7.12: Example of a **typical reconstruction result** where crown and trunk geometries are well aligned and visually realistic. The method produces coherent tree models that integrate smoothly with the urban context and are suitable as drag-imposing objects in CFD. (a) Raw ALS point cloud; (b) Reconstructed geometry with LoD 2.2 buildings from 3DBAG; (c) Crown hull projection showing centroid (blue) and municipal trunk location (red); (d) Segmented point cloud with individual tree clusters.

7.7. Answer to RQ3 and Future Work

The preceding analyses and results provide a direct answer to the third research question:

> *How can segmented tree point clouds be simplified into CFD-compatible geometries that balance geometric abstraction with drag characteristics for CFD?*

The results show that segmented tree clusters can be transformed into CFD-compatible geometries that capture their dominant drag effects in urban environments. Each tree is represented by a watertight crown mesh and a cylindrical trunk whose dimensions are estimated from crown geometry. Crown porosity is computed empirically per tree using a Monte Carlo-based characteristic gap size estimation followed by voxelisation, yielding robust values within the volumetric porosity range reported in wind tunnel experiments. This approach avoids voxel size and orientation sensitivity while remaining computationally feasible at city scale.

Overall, the reconstructed geometries provide a practical, stable, and aerodynamically meaningful representation of urban trees suitable for city-scale CFD. The reconstruction stage is computationally lightweight, as both the alpha wrapping and adaptive voxelisation procedures were tuned to minimise mesh complexity while preserving the required level of geometric realism. As a result, the full reconstruction, including porosity estimation and export, requires on average about one second per tree, enabling city-wide processing within practical time frames.

Future work could enhance this framework by (i) linking reconstructed crowns to municipal tree inventories to improve trunk placement realism, (ii) introducing species-dependent allometric relations for more accurate trunk dimensioning, (iii) partitioning crowns into subregions with locally varying porosity to capture intra-crown drag variation, (iv) validating the porosity estimates against empirical or high-resolution lidar measurements, and (v) improving segmentation accuracy to reduce over- and undersegmentation effects while maintaining scalability.

8. DISCUSSION AND CONCLUSIONS

8.1. Synthesis of Findings

This thesis set out to investigate whether urban trees can be automatically reconstructed from AHN airborne lidar into CFD-ready models that capture canopy structure and aerodynamic properties while remaining scalable. To this end, a complete pipeline was developed that integrates vegetation filtering, instance segmentation, taxonomic classification, and geometry reconstruction into a single automated workflow.

Segmentation (RQ1) and geometry reconstruction (RQ3) proved both feasible and consistent at city scale. On 16 CPU cores, the pipeline processed Amsterdam—the largest city in the Netherlands—in roughly 13 hours from raw point cloud to a 3D model and porosity estimates for about 380 000 detected trees across $\sim$240 km^2 of point cloud tiles. Tree crowns are abstracted into watertight alpha-wrapped geometries, trunks are estimated from canopy dimensions, and volumetric canopy porosity is quantified *per tree using an empirically calibrated voxelisation method* tuned to yield values within the 0.85–0.95 range reported in literature. Although geometry reconstruction and porosity estimation are computationally lightweight—requiring roughly one second per tree—they remain the main runtime component at city scale because processing scales linearly with the number of trees. These outputs form CFD-ready CityJSON vegetation objects.

Taxonomic classification (RQ2), however, proved unreliable with the available data and feature set. While the initial aim was to infer porosity from species or functional groups, the study showed that direct per-tree porosity estimation is a more viable approach at present. In this sense, the classification experiments clarified the limits of the current data rather than producing deployable models. The pipeline therefore prioritised breadth over depth: segmentation, classification, and geometry were all explored, but classification in particular would benefit from a dedicated follow-up study.

8.2. Cross-Cutting Discussion

The pipeline demonstrates how design choices in one stage propagate to others. Vegetation filtering removed very small trees, which was acceptable for CFD (they likely contribute negligible drag) but also removed potentially easy training samples, reducing classification performance. A similar effect arose from the minimum-point threshold in the reconstruction stage, where sparse tree point clouds (< 50 points) were excluded, further reducing the presence of small trees in the final dataset. Segmentation quality set an upper bound on the reliability of both classification and geometry: oversegmentation inflated trunk counts, while undersegmentation merged trees into bulk crowns that biased porosity. These segmentation artefacts also propagated computationally, as fragmented crowns increased the number of reconstructed geometries and thus overall processing time.

Similarly, the limited success of taxonomy-based classification redirected the workflow toward empirical, per-tree porosity estimation, strengthening the geometric branch of the pipeline while reducing dependence on incomplete inventory data.

Parameter tuning in the reconstruction stage reinforced this balance: the alpha-wrapping and voxelisation settings were chosen to maintain consistent connectivity and mesh simplicity across trees of varying size, ensuring both CFD stability and manageable runtime. Most design and tuning choices were further guided by the aim of generalisation, configuring the workflow to require as few user-defined parameters as possible. This enables it to operate robustly across diverse urban environments without requiring manual adjustments.

These interdependencies illustrate the main trade-off underpinning the design: the pipeline consistently prioritised robustness, scalability, and aerodynamic suitability over botanical realism. Geometries were deliberately simplified (watertight crowns, single bulk porosity values) to produce stable CFD inputs from noisy and incomplete data. This choice enabled city-scale automation but at the cost of fine biological detail. Taken together, these results show that while simplifications are inevitable, the reconstructed geometries provide a practical and aerodynamically meaningful representation of urban trees at city scale.

8.3. Conclusions and Future Work

This thesis presents the first end-to-end pipeline for reconstructing CFD-ready urban trees from unstructured airborne lidar at city scale. It integrates vegetation filtering, instance segmentation, and geometry reconstruction into a fully automated workflow that outputs CityJSON vegetation objects containing watertight crown meshes, trunk geometries, and empirically calibrated per-tree porosity values. The rotation- and offset-invariant porosity estimation method yields physically plausible volumetric values within reported literature ranges and can be used directly in CFD or other digital-twin applications. Parameter tuning across all stages was designed for generalisation and minimal user intervention, enabling scalable processing—on average about one second per tree—while maintaining geometric realism and aerodynamic relevance.

Segmentation quality remains the primary source of uncertainty, setting an upper bound on both classification and geometric accuracy. The taxonomy experiments showed that species-based porosity inference is currently limited by available data but clarified a viable alternative through direct empirical estimation. Overall, the workflow prioritised robustness, scalability, and CFD suitability over fine biological detail, demonstrating that simplified but coherent tree abstractions can meaningfully represent aerodynamic effects in urban simulations.

Future work should build on this foundation by improving segmentation to reduce over- and under-segmentation, integrating colour data (e.g. the upcoming AHN6 native RGB drape), and expanding labelled datasets to mitigate class imbalance, particularly for evergreen species. Further directions include incorporating species-dependent allometry for trunk dimensioning, refining trunk geometry (e.g. tapering or canopy-base-derived height), and modelling intra-crown porosity variation for enhanced drag representation. Additionally, although the current porosity values were tuned to align with the limited published range, empirical validation against wind-tunnel or field measurements would be valuable.

8.4. Closing Remarks

This work has shown that reconstructing CFD-ready trees from airborne lidar is both feasible and scalable. The complete implementation of the reconstruction pipeline developed for this thesis is released as open-source software, available under a GPL-3.0 licence at `https://github.com/NoahAlting/CFTree`. This repository, named `CFTree`, provides the full set of tools for data acquisition, vegetation segmentation, and geometric reconstruction up to the generation of CFD-ready tree models. It reflects the final operational workflow presented in this thesis and therefore excludes the experimental components related to taxonomic classification, which were explored separately but not integrated into the automated reconstruction process. The pipeline successfully processed Amsterdam, Rotterdam, Utrecht, and Delft, demonstrating that trees can now be represented explicitly in digital twins rather than being omitted or reduced to bulk porosity fields.

Urban trees play a vital role in real cities; with this capability, they can now claim their rightful place in digital twins.

A. APPENDIX: STUDY AREAS

This appendix shows the study areas used in this research. The shaded areas represent polygons derived by applying DBSCAN clustering and an alpha shape on municipal tree inventory points (shown as dots). These polygons were used to clip the AHN point clouds for each city, ensuring dense tree coverage while excluding isolated outliers. The background shows the AHN tile grid for spatial reference.

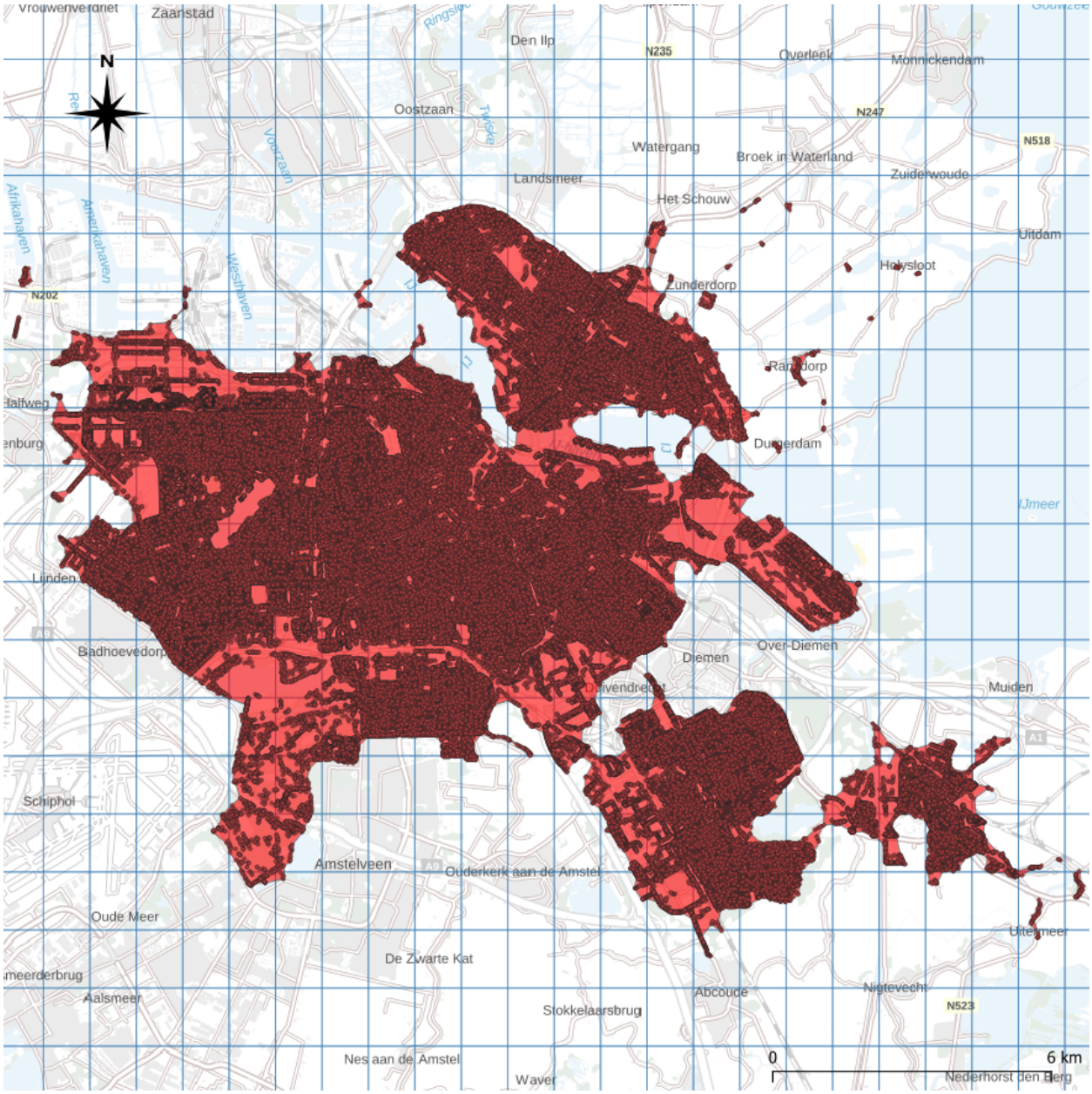


Figure A.1: Amsterdam study area with DBSCAN–alpha shape polygon (red) and municipal tree inventory points (dots), overlaid on the AHN tile grid.

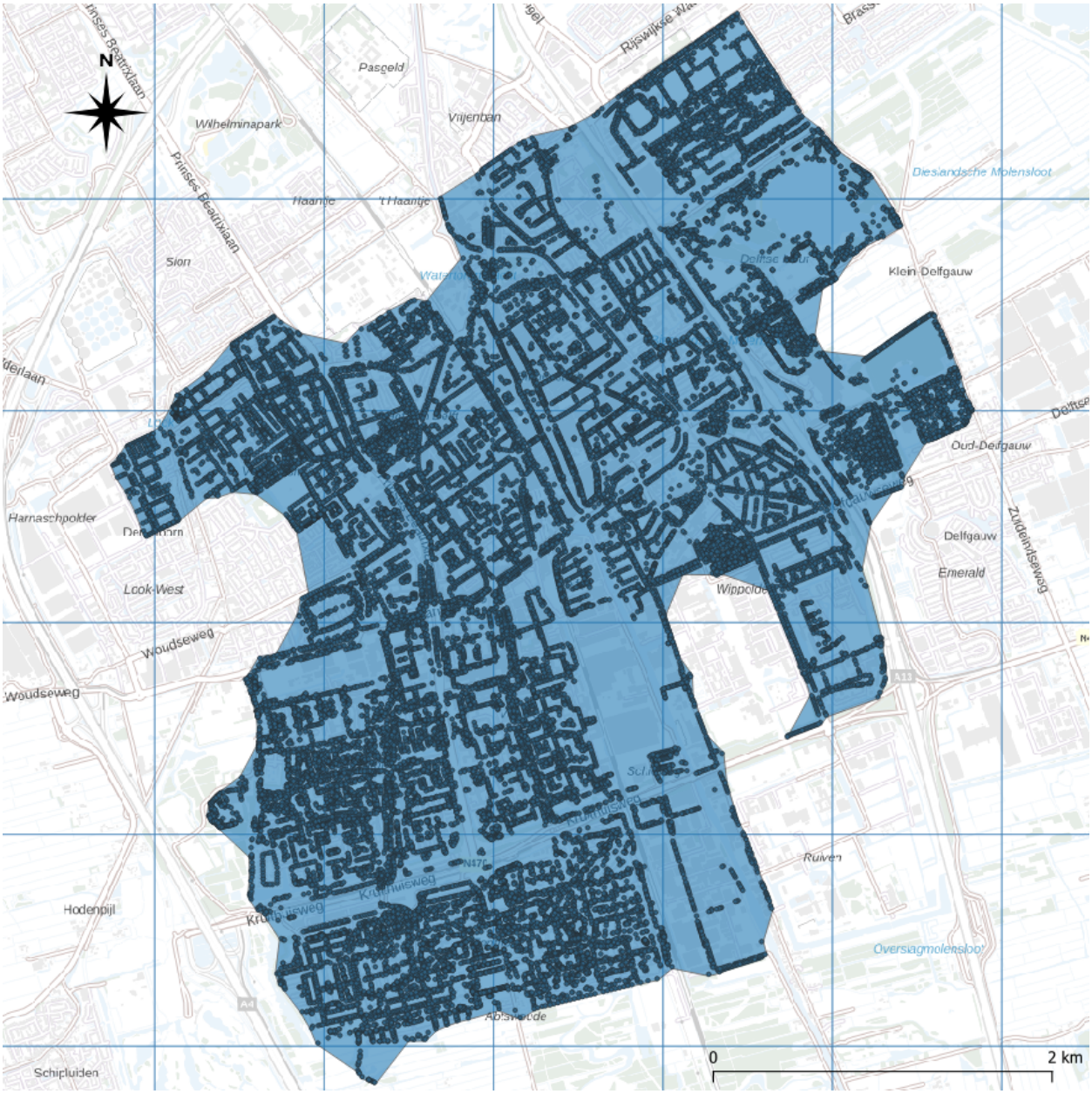


Figure A.2: Delft study area with DBSCAN-alpha shape polygon (blue) and municipal tree inventory points (dots), overlaid on the AHN tile grid.

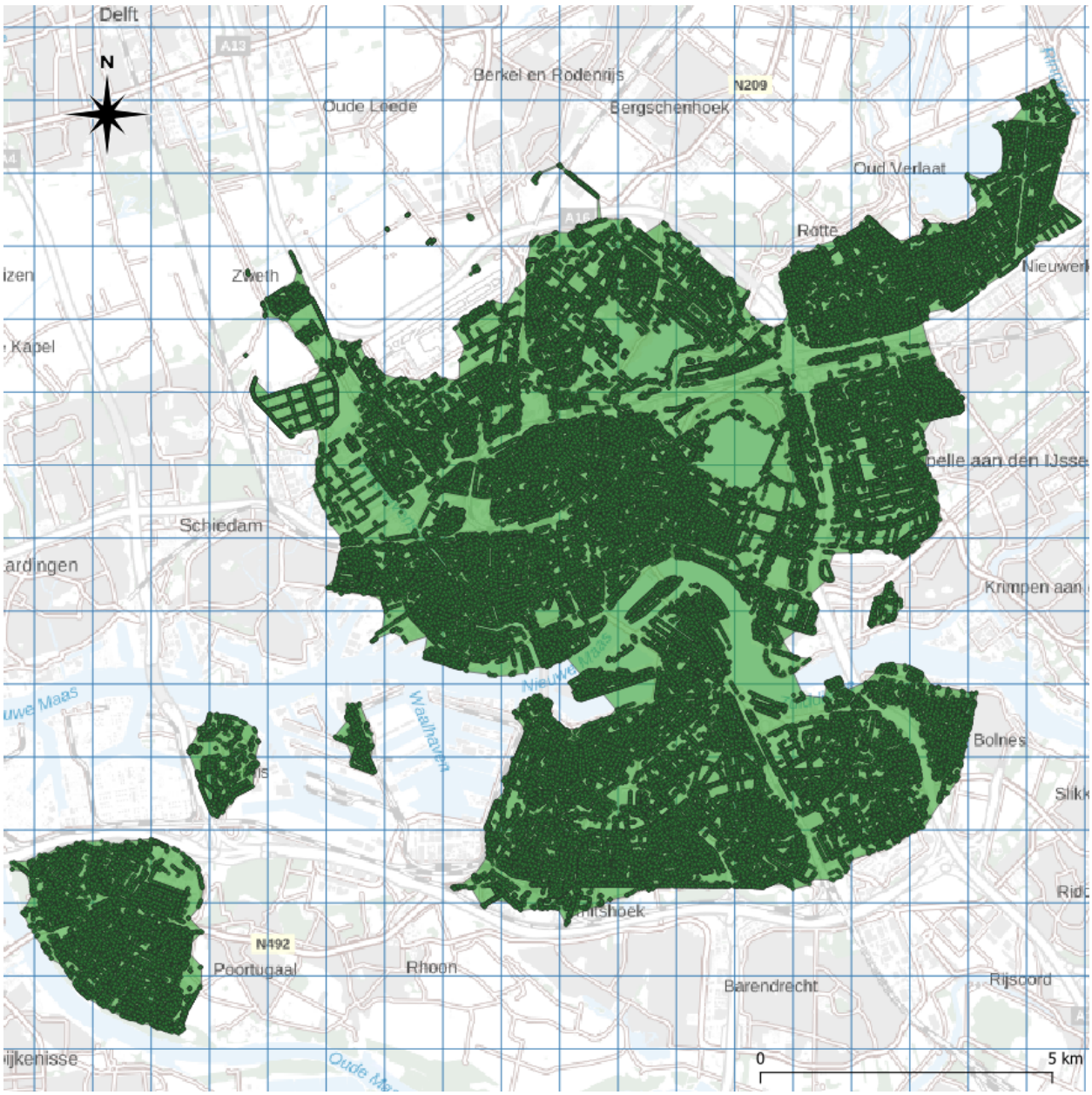


Figure A.3: Rotterdam study area (main part) with DBSCAN-alpha shape polygon (green) and municipal tree inventory points (dots), overlaid on the AHN tile grid.

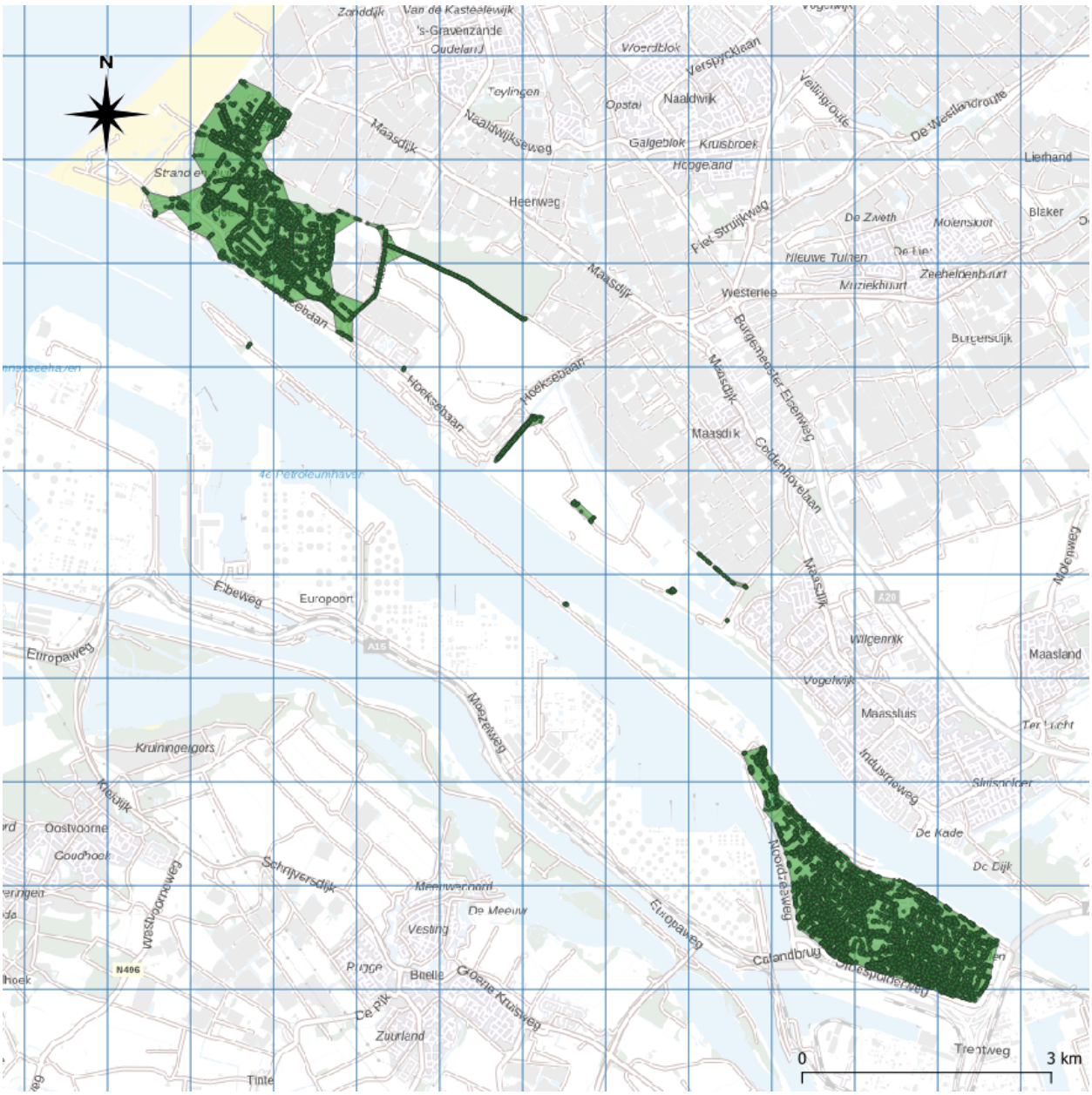


Figure A.4: Rotterdam study area (western part) with DBSCAN-alpha shape polygon (green) and municipal tree inventory points (dots), overlaid on the AHN tile grid.

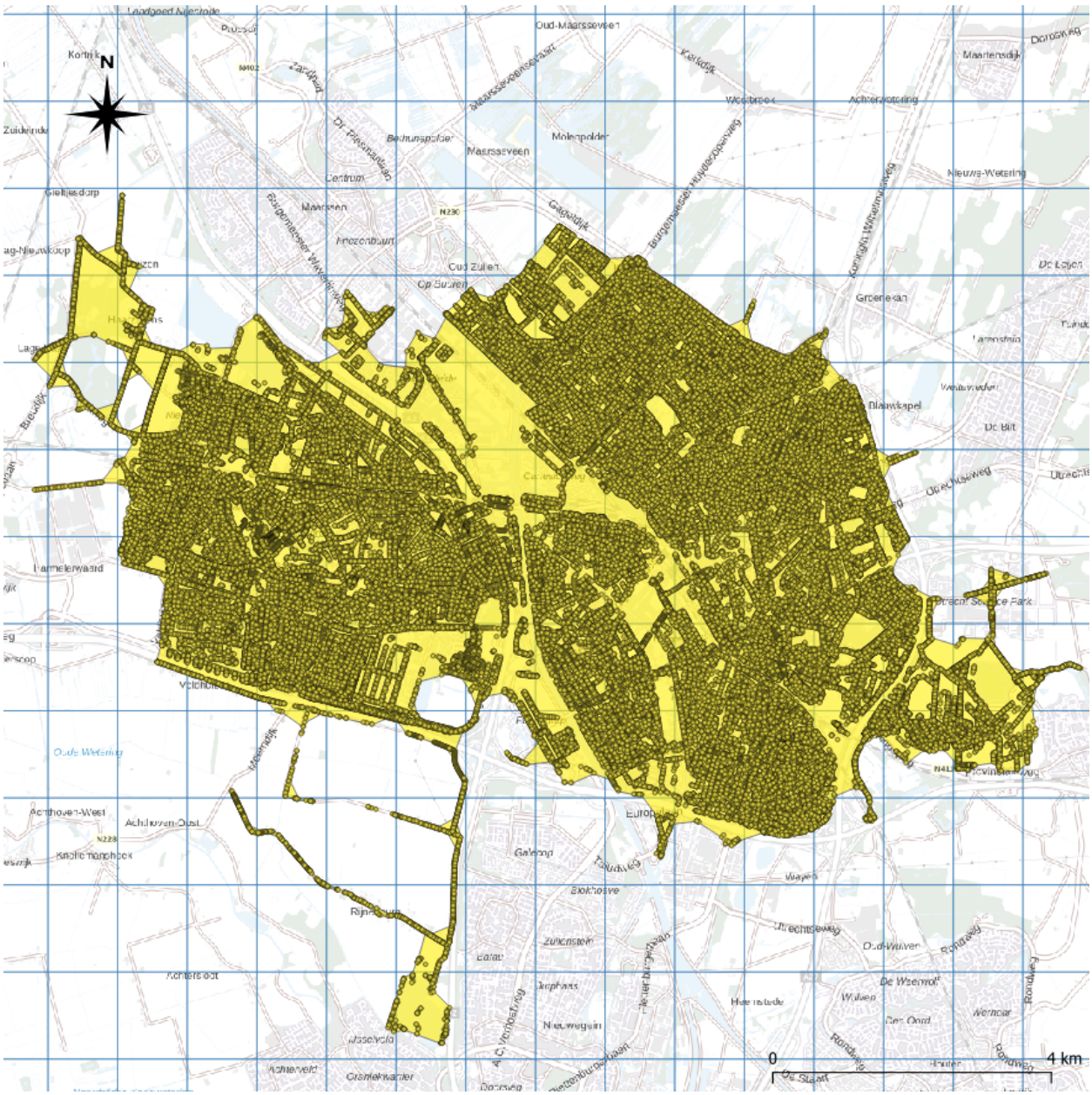


Figure A.5: Utrecht study area with DBSCAN-alpha shape polygon (yellow) and municipal tree inventory points (dots), overlaid on the AHN tile grid.

B. APPENDIX: SEGMENTATION IMPLEMENTATION AND RESULTS

This appendix provides supplementary material related to the tree instance segmentation. It first documents a minor implementation fix applied to the open-source *TreeSeparation* code used in this study, followed by the final segmentation results obtained in three representative neighbourhoods of Delft. The accompanying maps illustrate how the method partitions vegetation points into individual tree instances and highlight typical errors such as oversegmentation and undersegmentation. It also notes a small extension that enables the compiled binary to accept command-line input parameters for file paths and the main segmentation settings.

Implementation Notes and Minor Code Modifications

A minor implementation bug was identified in the `outputTrees()` function of the TreeSeparation repository by Wang et al. (2018), available at github.com/Jinhu-Wang/TreeSeparation. The original implementation (Code Snippet 1), located in `src/FoxTree.cpp` (around line 489), iterated over a `std::map<int, TreeCluster>` using sequential indices (`trees.at(i)`), which could lead to out-of-range access when map keys were not continuous. This loop was replaced by a range-based iteration over key–value pairs (Code Snippet 2) to ensure stable traversal of all clusters. Additionally, RGB colour assignment was removed, as colouring was not required in this workflow. These edits affect only the export routine and do not modify the segmentation logic or outcomes. In addition, a lightweight command-line interface was added to specify the input/output files and the main segmentation parameters at runtime, eliminating the need for recompilation during testing. Because these additions are generic and independent of the segmentation logic, they are not shown explicitly in the code listings below.

The original and revised implementations in `FoxTree.cpp` are listed below for reference.

Code Snippet 1: Original function `outputTrees()` from Wang et al. (2018) in `FoxTree.cpp`.

```
489 void FoxTree::outputTrees(std::string filename, std::map<int, TreeCluster>
        trees)
490 {
491     FILE* file = fopen(filename.c_str(), "w");
492 
493     for (int i = 0; i < trees.size(); ++i)
494     {
495         int r, g, b;
496         r = rand() % 255;
497         g = rand() % 255;
498         b = rand() % 255;
499 
500         for (int j = 0; j < trees.at(i).ptIDs.size(); ++j)
501         {
502             int id = trees.at(i).ptIDs.at(j);
503             Point3D pt = this->m_Points[id];
504             fprintf(file, "%d %lf %lf %lf %d %d %d\n",
505                 pt.treeID, pt.x, pt.y, pt.z, r, g, b);
506         }
507     }
508     fclose(file);
509 }
```

Code Snippet 2: Revised function `outputTrees()` correcting iteration and RGB handling in `FoxTree.cpp`.

```
489 void FoxTree::outputTrees(std::string filename, std::map<int, TreeCluster>
        trees)
490 {
491     FILE* file = fopen(filename.c_str(), "w");
492     int newTreeID = 0;
493 
494     for (const auto& [oldTreeID, cluster] : trees)
495     {
496         for (int id : cluster.ptIDs)
497         {
498             Point3D pt = this->m_Points[id];
499             fprintf(file, "%d %lf %lf %lf\n", newTreeID, pt.x, pt.y, pt.z)
                    ;
500         }
501         newTreeID++;
502     }
503 
504     fclose(file);
505 }
```

Segmentation Test Sites

Segmentation experiments were conducted on three neighbourhoods within Delft: Centrum Oost, Wippolder, and Vogelbuurt. These areas were selected to represent a variety of urban structures — from dense historic blocks (Centrum Oost) to mixed residential (Wippolder) and lower-density housing (Vogelbuurt). Their locations within the municipal boundary are shown below.

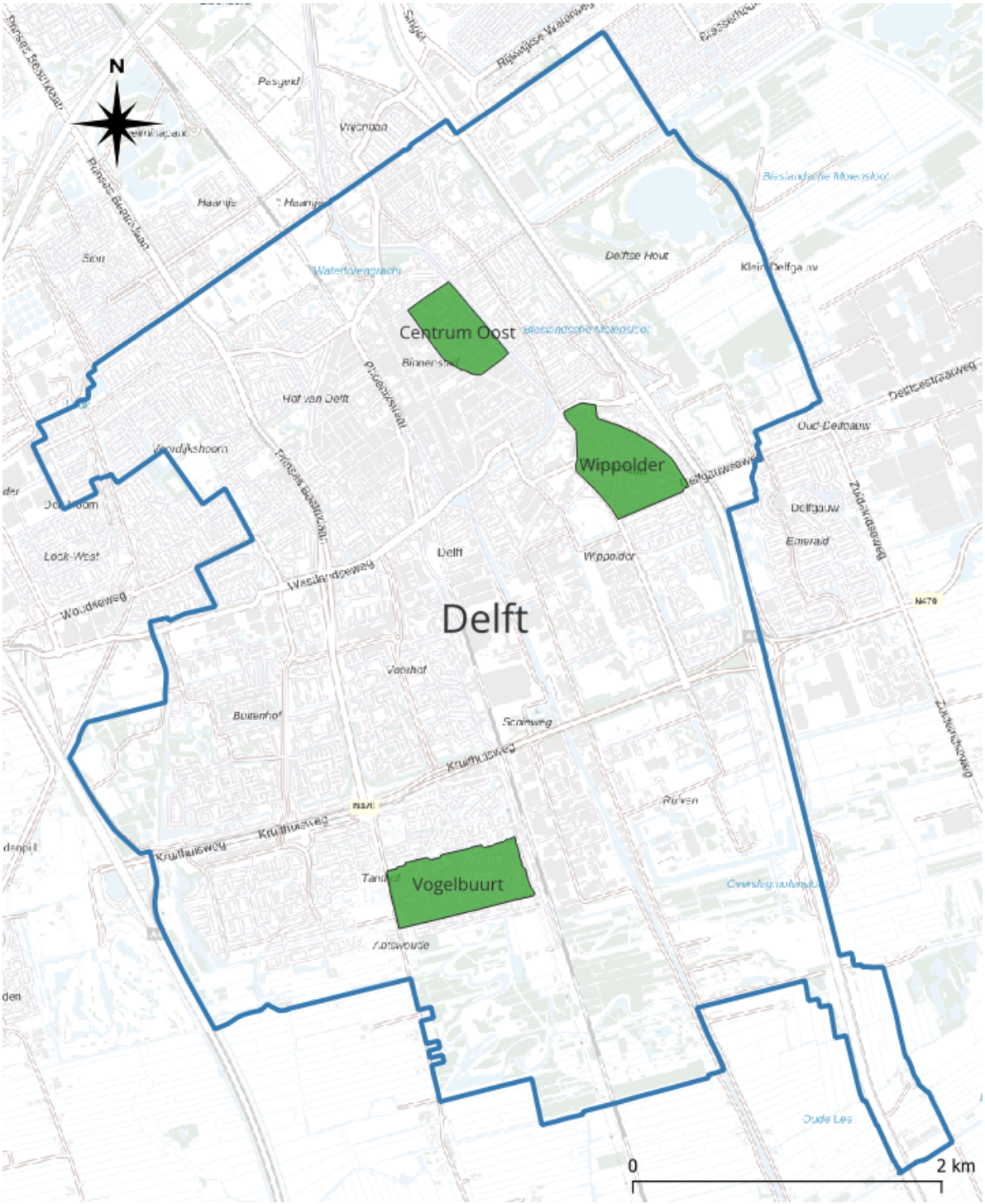


Figure B.1: Delft municipal boundary (blue) and the three neighbourhoods used for segmentation experiments (green).

Tuned Segmentation Results

Figure B.2: Segmentation hulls for Centrum Oost using $(r^*, v^*, m^*) = (2.5, 1.5, 3)$. Dark green = pure one-to-one hulls, light green = hulls with no registry point, red = oversegmented hulls (multiple hulls claiming the same point), orange = undersegmented hulls (one hull covering multiple registry points).

Figure B.3: Segmentation hulls for Vogelbuurt using $(r^*, v^*, m^*) = (2.5, 1.5, 3)$. Colour coding as in Figure B.2.

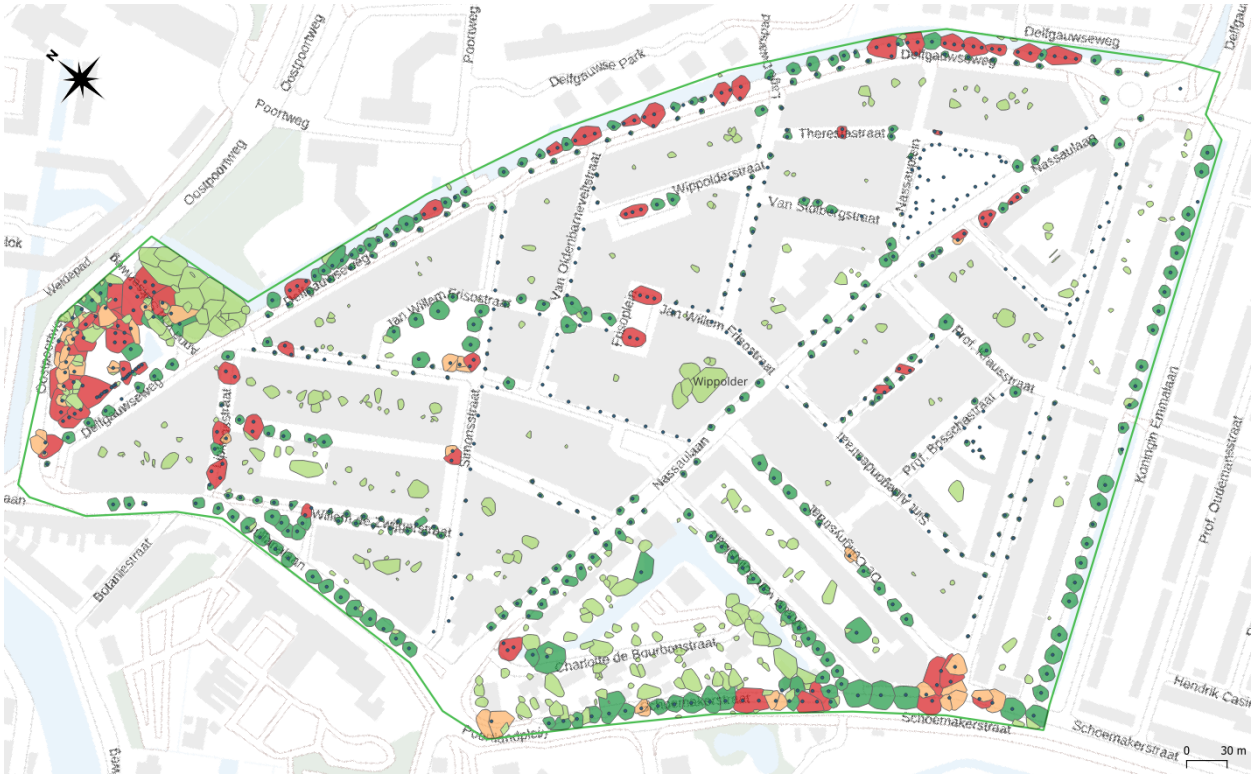


Figure B.4: Segmentation hulls for Wippolder using $(r^*, v^*, m^*) = (2.5, 1.5, 3)$. Colour coding as in Figure B.2.

C. APPENDIX: TREE POINT CLOUD FEATURES

This appendix lists the tree-level structural features described by Chi et al. (2025), which served as a reference for feature selection in this study.

Table C.1: Tree-level classification features and their descriptions as defined by Chi et al. (2025).

Feature	Description
Height	
Hmax, Hmed, Hmean	Maximum, Median and Mean height
Hbase	Crown base height
Hstd	Standard deviation of height
Hcv	Coefficient of variation of height
Hkur	Kurtosis of height
Hske	Skewness of height
Hp25, Hp90	25th, 90th percentile height
Hfirst_mean	Mean height of first-or-single returns
Intensity	
Imax, Imean	Maximum, Mean intensity
IaHmed	Mean intensity above median height
IbHmed	Mean intensity below median height
IabHmed	Ratio of IaHmed to IbHmed
Istd	Standard deviation of intensity
Icv	Coefficient of variation of intensity
Ikur	Kurtosis of intensity
Iske	Skewness of intensity
Ip25, Ip90	25th, 90th percentile intensity
Ifirst_mean	Mean intensity of first-or-single returns
Crown Size and Shape	
CWHmed	Crown width at median height
CWHp75, CWHp90	Crown width at 75th, 90th percentile height
CL_Hmax	Ratio of crown length to maximum height
Hmed_CW	Ratio of crown height to width: median height
Hp75_CW, Hp90_CW	Ratio of crown height to width: 75th, 90th percentile height
CWHp90_Hmean	Ratio of width at 90th percentile height to mean height
CWns_ew	Ratio of North-South width to East-West width
CRR	Canopy relief ratio
Crown Porosity and Density	
Hmean_med	(Hmean - Hmed) / Hmax
NHmean, NHmed	Return count in 0.5m vertical slice at mean, median height divided by width at that height
NHp90	Return count in 0.5 m vertical slice at 90th percentile height divided by width at that height
Nfirst	Percentage of first-or-single returns
Nlast	Percentage of last returns
N	Total number of points

D. APPENDIX: TAXONOMY RULES AND LABEL MAPPING

This appendix documents the taxonomy rules used in this thesis to assign biological labels to urban trees. The purpose of these rules is twofold: (i) to standardise the hierarchical taxonomic names found in municipal tree inventories (cultivar → species → genus → family), and (ii) to map all trees to a binary winter leaf-retention class (*evergreen* vs *deciduous*) for aerodynamic modelling.

All tables below are generated directly from the municipal tree inventories to ensure full reproducibility. The mapping is tailored to the datasets of Delft, Amsterdam, Utrecht and Rotterdam.

How to read the tables. Each row lists a family and its mapped genera. Genera are alphabetically ordered and grouped into multiple columns where needed. The evergreen/deciduous label is resolved using explicit overrides with precedence *species* > *genus* > *family*.

What each table shows:

- Table D.1 — Conifer families (default evergreen)
- Table D.2 — Non-conifer families
- Table D.3 — Explicit evergreen/deciduous overrides (species/genus level)
- Table D.4 — Alias mappings used during name normalisation
- Table D.5 — Placeholder values treated as unknowns

Table D.1: Conifer families and their mapped genera (default evergreen).

Family	Genera
Araucariaceae	Agathis Araucaria Wollemia
Cephalotaxaceae	Cephalotaxus
Cupressaceae	Calocedrus Chamaecyparis Cryptomeria Cunninghamia Cupressocyparis Cupressus Glyptostrobus Juniperus Metasequoia Platycladus Sequoia Sequoiadendron Taxodium Thuja Thujopsis
Ephedraceae	Ephedra
Gnetaceae	Gnetum
Pinaceae	Abies Cedrus Larix Picea Pinus Pseudolarix Pseudotsuga Tsuga
Podocarpaceae	Afrocarpus Dacrycarpus Dacrydium Podocarpus
Sciadopityaceae	Sciadopitys
Taxaceae	Taxus Torreya
Welwitschiaceae	Welwitschia

Table D.2: Non-conifer families and their mapped genera.

Family	Genera
Adoxaceae	Sambucus Viburnum
Altingiaceae	Liquidambar
Anacardiaceae	Cotinus Rhus Toxicodendron
Annonaceae	Asimina
Aquifoliaceae	Ilex
Araliaceae	Aralia Kalopanax Pseudopanax
Arecaceae	Trachycarpus
Asparagaceae	Ruscus
Berberidaceae	Berberis Mahonia Nandina

Family	Genera
Betulaceae	Alnus Betula Carpinus Corylus Ostrya
Bignoniaceae	Catalpa Chitalpa
Buxaceae	Buxus Sarcococca
Calycanthaceae	Chimonanthus
Cannabaceae	Celtis
Caprifoliaceae	Abelia Diervilla Dipelta Heptacodium Kolkwitzia Leycesteria Lonicera Weigela
Celastraceae	Euonymus
Cercidiphyllaceae	Cercidiphyllum
Clethraceae	Clethra
Cornaceae	Alangium Cornus
Ebenaceae	Diospyros
Elaeagnaceae	Elaeagnus Hippophae
Ericaceae	Arbutus Leucothoe Oxydendrum Rhododendron
Escalloniaceae	Escallonia
Eucommiaceae	Eucommia
Euphorbiaceae	Mallotus
Eupteleaceae	Euptelea
Fabaceae	Albizia Caragana Cercis Cladrastis Colutea Gleditsia Gymnocladus Laburnum Lespedeza Maackia Robinia Sophora Styphnolobium Wisteria
Fagaceae	Castanea Fagus Quercus
Garryaceae	Aucuba
Ginkgoaceae	Ginkgo
Hamamelidaceae	Corylopsis Disanthus Fothergilla Hamamelis Parrotia Parrotiopsis
Hydrangeaceae	Deutzia Hydrangea Philadelphus
Iteaceae	Itea
Juglandaceae	Carya Cyclocarya Juglans Pterocarya
Lamiaceae	Clerodendrum
Lardizabalaceae	Decaisnea
Lauraceae	Laurus Sassafras
Lythraceae	Lagerstroemia
Magnoliaceae	Liriodendron Magnolia
Malvaceae	Firmiana Hibiscus Tilia
Meliaceae	Cedrela Toona
Moraceae	Broussonetia Cudrania Ficus Maclura Morus
Myrtaceae	Eucalyptus
Nothofagaceae	Nothofagus
Nyssaceae	Davidia Nyssa
Oleaceae	Chionanthus Fraxinus Ligustrum Olea Osmanthus Syringa
Paulowniaceae	Paulownia
Platanaceae	Platanus
Poaceae	Koeleria
Rhamnaceae	Ceanothus Frangula Rhamnus Ziziphus
Rosaceae	Amelanchier Chaenomeles Cotoneaster Crataegus Cydonia Eriobotrya Exochorda Maddenia Malus Mespilus Oemleria Photinia Physocarpus Prunus Pyrus Rosa Sorbus
Rubiaceae	Emmenopterys
Rutaceae	Choisya Phellodendron Poncirus Ptelea Skimmia Tetradium Zanthoxylum
Sabiaceae	Meliosma

Family	Genera
Salicaceae	Idesia Populus Salix
Sapindaceae	Acer Aesculus Koelreuteria
Scrophulariaceae	Buddleja
Simaroubaceae	Ailanthus Picrasma
Staphyleaceae	Staphylea
Styracaceae	Halesia Pterostyrax Styrax
Symplocaceae	Symplocos
Tamaricaceae	Tamarix
Theaceae	Stewartia
Thymelaeaceae	Edgeworthia
Trochodendraceae	Tetracentron
Ulmaceae	Planera Ulmus Zelkova
Vitaceae	Parthenocissus

Table D.3: Explicit label overrides used for evergreen/deciduous assignment.

Level	Genus	Epithet	Assigned label	Note
genus	Glyptostrobus		deciduous	deciduous conifer override
genus	Larix		deciduous	deciduous conifer override
genus	Metasequoia		deciduous	deciduous conifer override
genus	Taxodium		deciduous	deciduous conifer override
species	Larix	decidua	deciduous	deciduous conifer override
species	Metasequoia	glyptostroboides	deciduous	deciduous conifer override
species	Taxodium	distichum	deciduous	deciduous conifer override
genus	Buxus		evergreen	broadleaf evergreen override
genus	Eucalyptus		evergreen	broadleaf evergreen override
genus	Ilex		evergreen	broadleaf evergreen override
genus	Laurus		evergreen	broadleaf evergreen override
genus	Osmanthus		evergreen	broadleaf evergreen override
genus	Sarcococca		evergreen	broadleaf evergreen override
species	Eriobotrya	japonica	evergreen	broadleaf evergreen override
species	Ilex	aquifolium	evergreen	broadleaf evergreen override
species	Laurus	nobilis	evergreen	broadleaf evergreen override
species	Ligustrum	ovalifolium	evergreen	broadleaf evergreen override
species	Magnolia	grandiflora	evergreen	broadleaf evergreen override
species	Photinia	davidiana	evergreen	broadleaf evergreen override
species	Photinia	fraseri	evergreen	broadleaf evergreen override
species	Prunus	laurocerasus	evergreen	broadleaf evergreen override
species	Quercus	ilex	evergreen	broadleaf evergreen override
species	Ilex	verticillata	deciduous	broadleaf deciduous override

Table D.4: Alias mappings applied during taxon normalisation.

Type	From	To
taxon_alias	chinese vernisboom	Toxicodendron vernicifluum
taxon_alias	noordelijke japanse magnolia	Magnolia kobus
taxon_alias	okkernoot	Juglans
genus_alias	Crataemespilus	Crataegus
genus_alias	Metasquoia	Metasequoia
genus_alias	Okkernoot	Juglans

Type	From	To
genus_alias	Pterocayra	Pterocarya
genus_alias	Qercus	Quercus
genus_alias	Seringa	Syringa
genus_alias	Sycoparrotia	Parrotia
genus_alias	Tilla	Tilia

Table D.5: Placeholders treated as unknowns during taxon normalisation.

Type	Value
taxon_placeholder	assortiment
taxon_placeholder	nader te bepalen
taxon_placeholder	onbekend
taxon_placeholder	overig
genus_placeholder	Coniferen
genus_placeholder	Niet
genus_placeholder	Nvt
genus_placeholder	Onbekend
genus_placeholder	Overig
genus_placeholder	assortiment
species_placeholder	soort
species_placeholder	sp
species_placeholder	sp.
species_placeholder	spec
species_placeholder	spec.
species_placeholder	species
species_placeholder	spp
species_placeholder	spp.

E. APPENDIX: TAXONOMIC CLASSIFICATION RESULTS

This appendix summarises the Random Forest (RF) experiments conducted to classify trees from point cloud features. It provides (i) the hyperparameter search space, (ii) the configuration and metrics of the best-performing models, and (iii) confusion matrices to visualise class-level prediction performance.

Hyperparameter Tuning Details

The table below lists the RF hyperparameters tuned via a deterministic grid search with 5-fold cross-validation. To reduce computational cost, the number of estimators was fixed to 200 during cross-validation and increased for the final refit.

Table E.1: Random Forest hyperparameters tuned via a deterministic grid and 5-fold cross-validation (CV). During cross-validation, `n_estimators` is fixed at 200 to reduce computation; the best configuration was then refit with `n_estimators`=800 on all training data.

Hyperparameter	**Grid values**	**Explanation**
`n_estimators`	CV: 200; Final: 800	Number of trees. More trees reduce variance and stabilise performance at the cost of training time and model size.
`max_depth`	`{None, 16, 24, 32}`	Maximum tree depth. Caps model complexity; smaller values reduce overfitting.
`min_samples_leaf`	`{1, 3, 7}`	Minimum samples required at a leaf node. Larger values smooth decision boundaries.
`min_samples_split`	`{2, 10, 30}`	Minimum samples required to split an internal node.
`max_features`	`{sqrt, log2, 30% of features}`	Features considered at each split. Lower values increase tree diversity.
`criterion`	`{gini, entropy}`	Impurity measure for split evaluation.
`max_samples` (bootstrap fraction)	`{0.7, 1.0}`	Fraction of training rows (with replacement) drawn to build each tree when `bootstrap=True`.

Best Performing Model Details

The table below summarises dataset coverage after class balancing, the best hyperparameters found for each taxonomic rank, and the main evaluation metrics on the held-out test set.

Table E.2: Coverage diagnostics, best hyperparameters, and quick-run metrics.

Metric	Species	Genus	Family	Leaf type
Rare cutoff (min samples)	50	50	50	50
Rare classes dropped (<50 samples)	325	69	23	0
Classes left after rare-drop	147	57	28	2
Kept head classes (95% coverage)	79	25	12	1
Final classes (incl. Other)	80	26	13	2
Train samples	190 841	206 902	206 750	206 912
Test samples	47 711	51 726	51 688	51 728
n_estimators	480	480	480	240
min_samples_leaf	1	1	1	3
min_samples_split	10	10	10	10
max_features	sqrt	sqrt	sqrt	sqrt
criterion	gini	gini	gini	gini
Macro-F1 (all classes)	0.270	0.372	0.442	0.722
Macro-F1 (head only)	0.276	0.374	0.437	0.500
Balanced accuracy	0.289	0.397	0.442	0.681
Micro-F1	0.356	0.439	0.496	0.995

Confusion Matrices

The confusion matrices below illustrate class-level prediction behaviour for each classifier. Correct predictions appear along the diagonal, while off-diagonal cells indicate misclassifications. Sharper diagonals reflect stronger class separation (as seen for leaf_type), whereas dispersed patterns reflect confusion between classes (as seen for species). Per-cell accuracy values are shown for leaf_type and family, but omitted for genus and species because the large number of classes would make them unreadable.

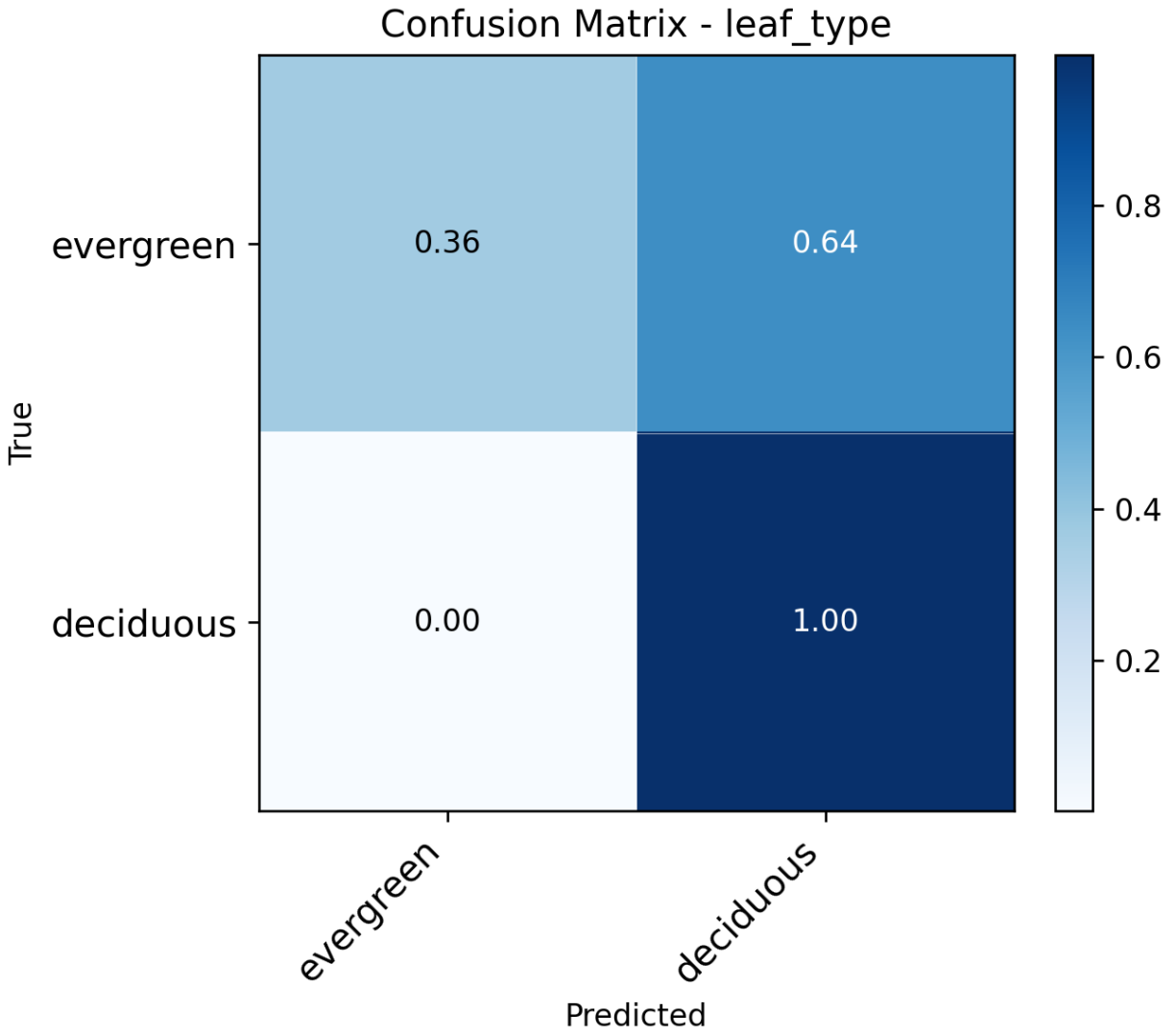


Figure E.1: Confusion matrix for the leaf_type classifier (evergreen vs deciduous). Predictions are strongly skewed toward the majority deciduous class, with few evergreen trees correctly detected.

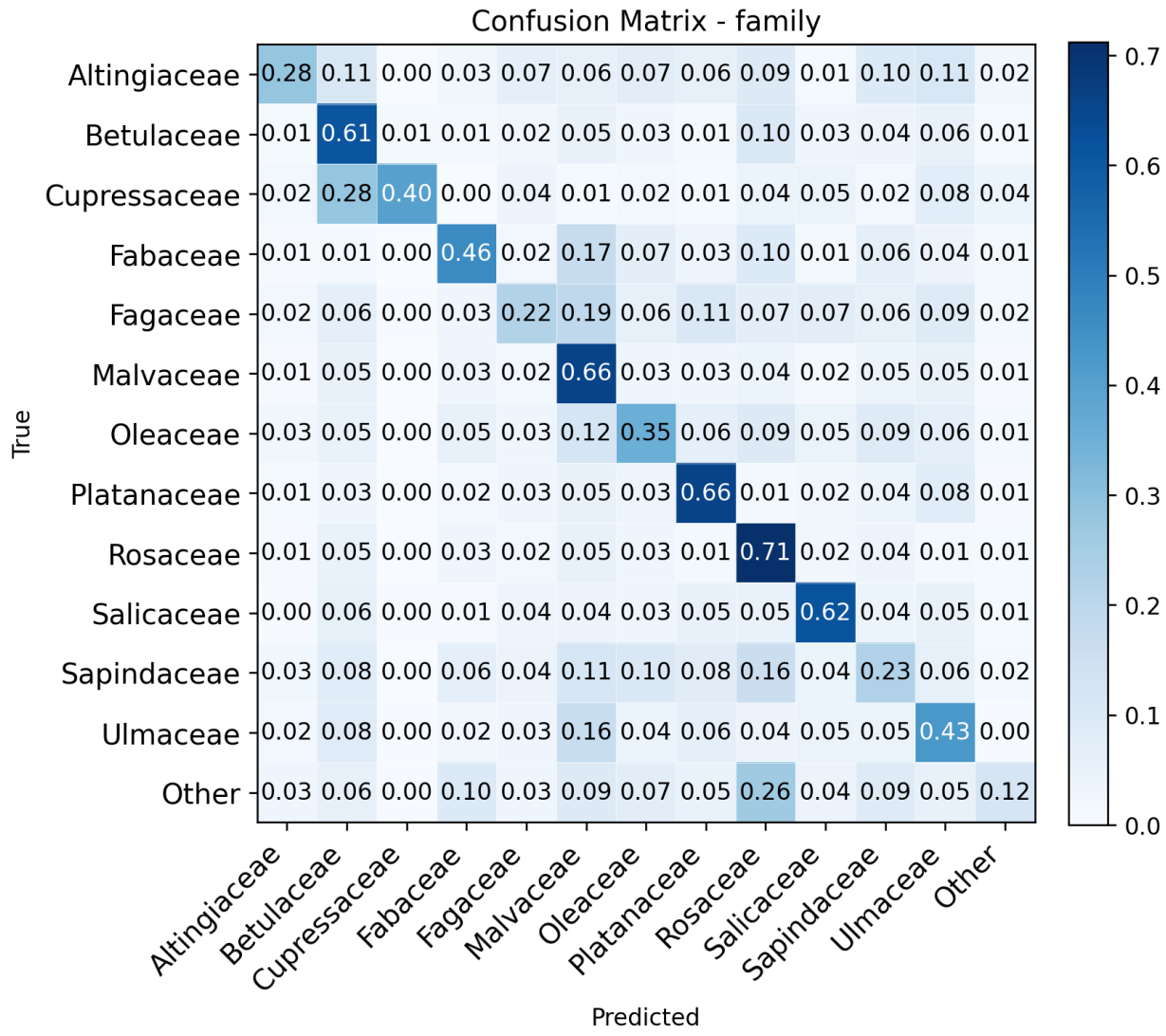


Figure E.2: Confusion matrix for the family-level classifier. A few dominant families show modest clustering along the diagonal, but most families are frequently confused.

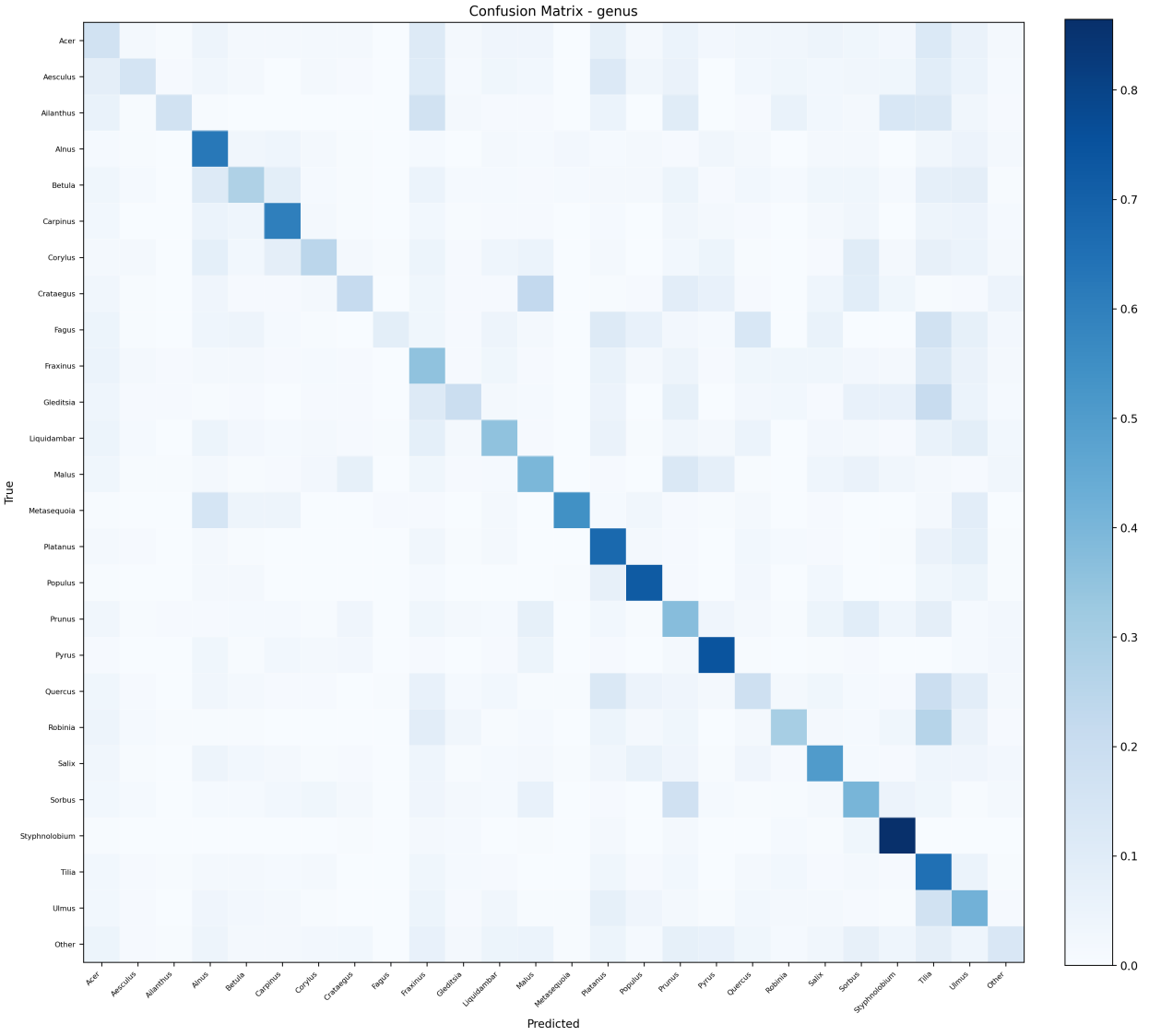


Figure E.3: Confusion matrix for the genus-level classifier. Predictions show partial clustering along the diagonal but substantial confusion across genera. Per-cell values are omitted for readability due to the large number of classes.

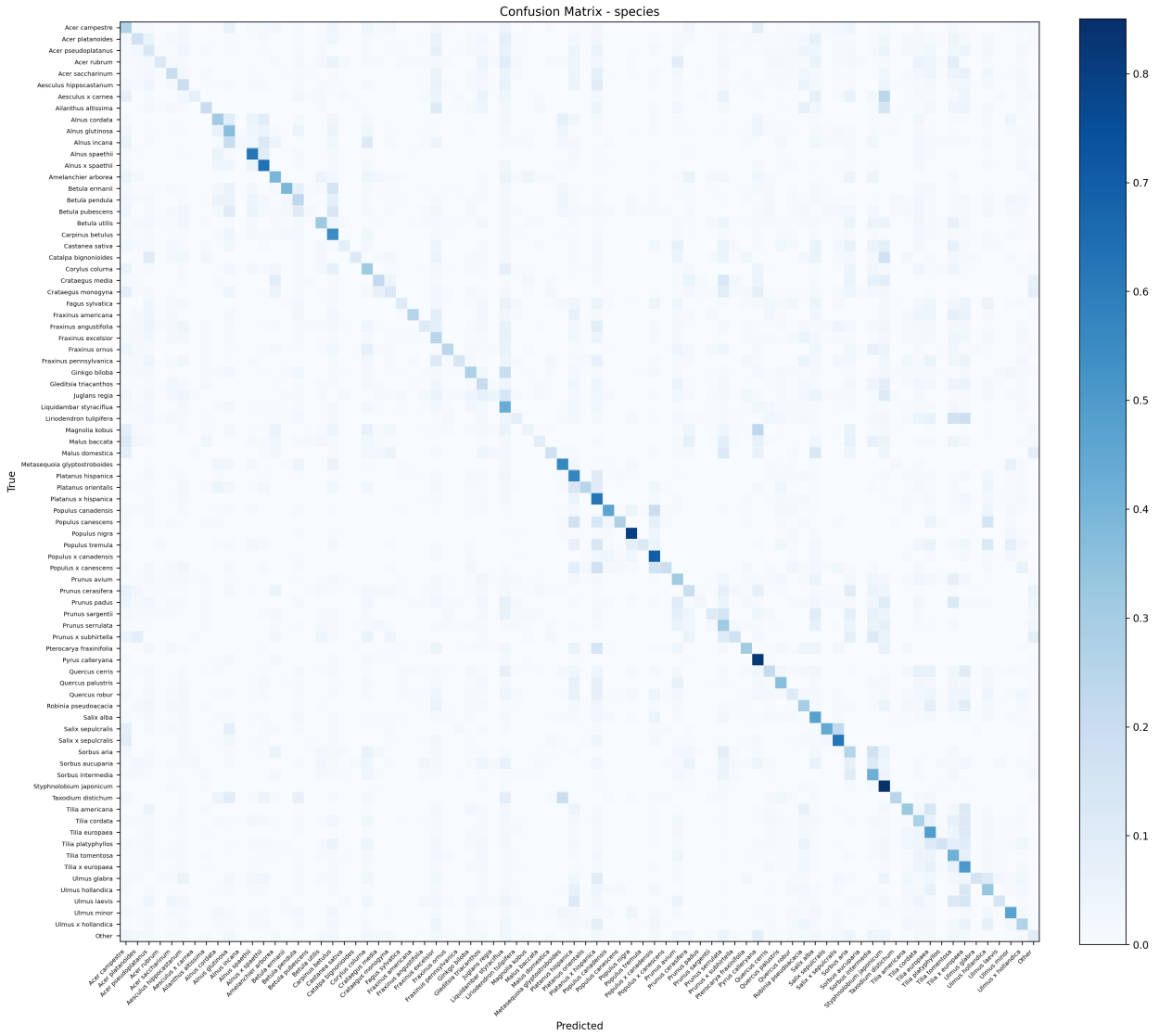


Figure E.4: Confusion matrix for the species-level classifier. Predictions are widely dispersed, with only a few species showing distinct clusters along the diagonal. Per-cell values are omitted for readability due to the large number of classes.

REFERENCES

Alliez, P., Cohen-Steiner, D., Hemmer, M., Portaneri, C., and Rouxel-Labbé, M. (2025). 3D alpha wrapping. In *CGAL User and Reference Manual*. CGAL Editorial Board, 6.1 edition.

Amiri, N., Krzystek, P., Heurich, M., and Skidmore, A. (2019). Classification of tree species as well as standing dead trees using triple wavelength als in a temperate forest. *Remote Sensing*, 11(22):2614.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Čeperković, A. (2015). Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889.

Bitog, J. P., Lee, I.-B., Hwang, H.-S., Shin, M.-H., Hong, S.-W., Seo, I.-H., Mostafa, E., and Pang, Z. (2011). A wind tunnel study on aerodynamic porosity and windbreak drag. *Forest Science and Technology*, 7(1):8–16.

Brandle, J. R. (2009). Windbreak practices. In Garrett, H. E., editor, *North American Agroforestry: An Integrated Science and Practice*, chapter 5, pages 75–104. American Society of Agronomy, Madison, WI, 2 edition.

Brickell, C. D., Alexander, C., Cubey, J. J., David, J. C., Hoffmann, M. H. A., and et al., editors (2016). *International Code of Nomenclature for Cultivated Plants (ICNCP), Ninth Edition*, volume 18 of *Scripta Horticulturae*. International Society for Horticultural Science, Leuven.

Buccolieri, R., Santiago, J.-L., Rivas, E., and Sanchez, B. (2018). Review on urban tree modelling in CFD simulations: Aerodynamic, deposition and thermal effects. *Urban Forestry & Urban Greening*, 31:212–220.

Chehreh, B., Moutinho, A., and Viegas, C. (2023). Latest trends on tree classification and segmentation using uav data—a review of agroforestry applications. *Remote Sensing*, 15(9):2263.

Chen, J., Chen, Y., and Liu, Z. (2021a). Classification of typical tree species in laser point cloud based on deep learning. *Remote Sensing*, 13(23):4750.

Chen, Q., Baldocchi, D., Gong, P., and Kelly, M. (2006). Isolating individual trees in a savanna woodland using small footprint lidar data. *Photogrammetric Engineering & Remote Sensing*, 72(8):923–932.

Chen, X., Jiang, K., Zhu, Y., Wang, X., and Yun, T. (2021b). Individual tree crown segmentation directly from uav-borne lidar data using the pointnet of deep learning. *Forests*, 12(2):131.

Chi, D., Yan, J., Yu, K., Morsdorf, F., and Somers, B. (2025). Planting contexts affect urban tree species classification using airborne hyperspectral and lidar imagery. *Landscape and Urban Planning*, 257:105316.

de Groot, G. J. (2020). Automatic construction of 3d tree models in multiple levels of detail from airborne lidar data. Master's thesis, TU Delft.

Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72.

Fu, L., Duan, G., Ye, Q., Meng, X., Luo, P., Sharma, R. P., Sun, H., Wang, G., and Liu, Q. (2020). Prediction of individual tree diameter using a nonlinear mixed-effects modeling approach and airborne lidar data. *Remote Sensing*, 12(7):1066.

Fu, R., Pađen, I., and García-Sánchez, C. (2024). Should we care about the level of detail in trees when running urban microscale simulations? *Sustainable Cities and Society*, 101:105143.

García-Sánchez, C., Vitalis, S., Pađen, I., and Stoter, J. (2021). The impact of level of detail in 3d city models for cfd-based wind flow simulations. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-4-W4-2021:67–72. ISPRS TC IV – 16th 3D GeoInfo Conference 2021, New York City, USA.

Gröger, G. and Plümer, L. (2012). Citygml – interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33.

Haboudane, D., Miller, J. R., Pattey, E., Zarco-Tejada, P. J., and Strachan, I. B. (2004). Hyperspectral vegetation indices and novel algorithms for predicting green lai of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 90(3):337–352.

Hell, M., Brandmeier, M., Briechle, S., and Krzystek, P. (2022). Classification of tree species and standing dead trees with lidar point clouds using two deep neural networks: Pointcnn and 3dmfv-net. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 90(2):103–121.

Hermann, T. (2024). Leaf it to AI: Mapping urban tree morphology and leaf area index with multimodal deep-learning. Master's thesis, Ecole Polytechnique Federale de Lausanne.

Hong, B., Lin, B., and Qin, H. (2017). Numerical investigation on the effect of avenue trees on pm2.5 dispersion in urban street canyons. *Atmosphere*, 8(7):129.

Hu, X., Chen, W., and Xu, W. (2017). Adaptive mean shift-based identification of individual trees using airborne lidar data. *Remote Sensing*, 9(2):148.

Jaiswal, M., Corpuz, A. M., and Hsu, M.-C. (2024). Mesh-driven resampling and regularization for robust point cloud-based flow analysis directly on scanned objects. *Computer Methods in Applied Mechanics and Engineering*, 432:117426.

Jakubowski, M. K., Li, W., Guo, Q., and Kelly, M. (2013). Delineating individual trees from lidar data: A comparison of vector- and raster-based segmentation approaches. *Remote Sensing*, 5(9):4163–4186.

Jeanjean, A., Monks, P., and Leigh, R. (2016). Modelling the effectiveness of urban trees and grass on pm2.5 reduction via dispersion and deposition at a city scale. *Atmospheric Environment*, 147.

Kamoske, A. G., Dahlin, K. M., Stark, S. C., and Serbin, S. P. (2019). Leaf area density from airborne LiDAR: Comparing sensors and resolutions in a temperate broadleaf forest ecosystem. *Forest Ecology and Management*, 433:364–375.

Keerthinathan, P., Winsen, M., Krishnakumar, T., Ariyanayagam, A., Hamilton, G., and Gonzalez, F. (2025). Modelling lidar-based vegetation geometry for computational fluid dynamics heat transfer models. *Remote Sensing*, 17(3):552.

Lee, J., Coomes, D., Schonlieb, C.-B., Cai, X., Lellmann, J., Dalponte, M., Malhi, Y., Butt, N., and Morecroft, M. (2017). A graph cut approach to 3D tree delineation, using integrated airborne LiDAR and hyperspectral imagery. arXiv:1701.06715 [cs.CV].

Lee, S.-H., Kim, H., Moon, H., Kim, H.-S., Han, S.-S., and Jeong, S. (2023). Effects of wind barrier porosity and inclination on wind speed reduction. *Applied Sciences*, 13(14):8310.

Mao, Z., Lu, Z., Wu, Y., and Deng, L. (2023). DBH estimation for individual tree: Two-dimensional images or three-dimensional point clouds? *Remote Sensing*, 15(16):4116.

Neyns, S. and Canters, F. (2022). Mapping of urban vegetation with high-resolution remote sensing: A review. *Remote Sensing*, 14(3):629.

Nowak, M. M., Pedziwiatr, K., and Bogawski, P. (2022). Hidden gaps under the canopy: Lidar-based detection and quantification of porosity in tree belts. *Ecological Indicators*, 142:109243.

Ortega-Córdova, L. M. (2018). Urban vegetation modeling in 3d levels of detail. Master's thesis, TU Delft.

Parker, G. G. (2020). Tamm review: Leaf area index (lai) is both a determinant and a consequence of important processes in vegetation canopies. *Forest Ecology and Management*, 477:118496.

Pađen, I., García-Sánchez, C., and Ledoux, H. (2022). Towards automatic reconstruction of 3d city models tailored for urban flow simulations. *Frontiers in Built Environment*, 8.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., and Stoter, J. (2022). Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands. *Photogrammetric Engineering and Remote Sensing*, 88(3):165–170.

Pfeiffer, S. A., Guevara, J., Cheein, F. A., and Sanz, R. (2018). Mechatronic terrestrial lidar for canopy porosity and crown surface estimation. *Computers and Electronics in Agriculture*, 146:104–113.

Popescu, S. C., Wynne, R. H., and Nelson, R. F. (2003). Measuring individual tree crown diameter with lidar and assessing its influence on estimating forest volume and biomass. *Canadian Journal of Remote Sensing*, 29(5):564–577.

Popov, M., Semko, I., Kozak, I., and Kozlova, A. (2024). Improved method to calculate urban forest vertical structure using airborne laser scanning data. In Zagorodny, A., Bogdanov, V., and Zaporozhets, A., editors, *Nexus of Sustainability: Understanding of FEWSE Systems I*, pages 295–314. Springer Nature Switzerland, Cham.

Reitberger, J., Schnörr, C., Krzystek, P., and Stilla, U. (2009). 3d segmentation of single trees exploiting full waveform lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):561–574.

Ricci, A., Kalkman, I., Blocken, B., Burlando, M., Freda, A., and Repetto, M. P. (2017). Local-scale forcing effects on wind flows in an urban environment: Impact of geometrical simplifications. *Journal of Wind Engineering and Industrial Aerodynamics*, 170:238–255.

Rodriguez, A., Lecigne, B., Wood, S., Carmeliet, J., Kubilay, A., and Derome, D. (2024). Optimal representation of tree foliage for local urban climate modeling. *Sustainable Cities and Society*, 115:105857.

Rusu, R., Marton, Z., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56:927–941.

Slavík, M., Kuželka, K., Modlinger, R., and Surový, P. (2023). Spatial analysis of dense lidar point clouds for tree species group classification using individual tree metrics. *Forests*, 14(8):1581.

Somanath, S., Naserentin, V., Eleftheriou, O., Sjölie, D., Wästberg, B. S., and Logg, A. (2024). Towards urban digital twins: A workflow for procedural visualization using geospatial data. *Remote Sensing*, 16(11):1939.

Stuurgroep AHN (2023). Kwaliteitsbeschrijving Actueel Hoogtebestand Nederland (AHN). https://www.ahn.nl/kwaliteitsbeschrijving. Accessed: October 2025.

The Angiosperm Phylogeny Group (2016). An update of the angiosperm phylogeny group classification for the orders and families of flowering plants: Apg iv. *Botanical Journal of the Linnean Society*, 181(1):1–20.

The World Bank (2023). Urban population (% of total population) - netherlands (1960–2023). Accessed: October 2025.

Turland, N. J., Wiersema, J. H., Barrie, F. R., Greuter, W., Hawksworth, D. L., and et al., editors (2018). *International Code of Nomenclature for Algae, Fungi, and Plants (Shenzhen Code) adopted by the Nineteenth International Botanical Congress Shenzhen, China, July 2017*, volume 159 of *Regnum Vegetabile*. Koeltz Botanical Books, Glashütten.

van Natijne, A., of Geoscience, D., and Remote Sensing, T. D. (2023). Geotiles: readymade geodata with a focus on the Netherlands. https://geotiles.citg.tudelft.nl/. CC BY 4.0 licence.

Wang, J., Lindenbergh, R., and Menenti, M. (2018). Scalable individual tree delineation in 3D point clouds. *The Photogrammetric Record*, 33(163):315–340.

Weinstein, B. G., Marconi, S., Aubry-Kientz, M., Vincent, G., Senyondo, H., and White, E. P. (2020). DeepForest: A python package for RGB deep learning tree crown delineation. *Methods in Ecology and Evolution*, 11(12):1743–1751.

Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., and Liu, H. (2013). A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sensing*, 5(2):584–611.

Zhang, C., Zhou, Y., and Qiu, F. (2015). Individual tree segmentation from lidar point clouds for urban forest inventory. *Remote Sensing*, 7(6):7892–7913.