MSc thesis in Geomatics

Assessing The Impact of Tree 3D Representations on Urban Daylight Simulation Based on Airborne Laser Scanning Point Cloud Data

Victoria Tsalapati

2025

TU Delft Delft University of Technology

MSc thesis in Geomatics

Assessing The Impact of Tree 3D Representations on Urban Daylight Simulation Based on Airborne Laser Scanning Point Cloud Data

Victoria Tsalapati

October 2025

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

Geo-Database Management Centre
Delft University of Technology

Supervisors:	Dr. A. Rafiee
	Dr. E. Brembilla
Co-reader:	Ir. E. Verbree

Abstract

Daylight is a significant factor in the decision-making process of urban planners and architects when intervening in an urban environment. A considerable element in such a complex environment, assessed in daylight simulations, is greenery, particularly trees, due to their interaction with sunlight and their shading effects.
This thesis investigates the impact of diverse tree 3D representations produced by Actueel Hoogtebestand Nederland (AHN) point cloud data when performing daylight simulation on scenes integrating them. The assessed tree 3D representations are point cloud based representation using small cubes instead of the points from the original point cloud, voxels, alpha shapes of individual trees, and convex hull of branch points. The tool for the daylight simulation was the Daylight Availability Workflow of Climate Studio plugin in Rhino software which produces time series of simulated illuminance values based on the location and the local solar data, the scene, the material properties of the objects of the scene and surface on which simulation is performed. Specifically, regarding the assigned materials for the voxel presentation, two approaches were followed; one related to the predefined opaque material properties in Rhino software and the other to transparent materials defined by the density point inside a voxel. The reference of the simulations was a dataset of actual illuminance values recorded by the sensor located on the west facade of the CCC building, where is the position that the simulation was conducted. Also, to examine how the seasonal changes influence the simulation results, simulations for two months, February and June, were performed. For the June simulations, a synthesized point cloud was generated by combining the AHN points with additional points representing the tree canopy.
Next, results from all days in both months the results showed that the point cloud based representation caused a significant overestimation of simulated illuminance values, whereas the Alpha Shape and Convex Hull representations resulted in underestimation. In contrast, the simulation outcomes for voxel representations depended on their material properties (opaque or transparent), spatial allocation, and size. It was proven that for February via the voxel representations of sizes close to 0.10 m there was the best fit between simulations and sensor data, yet it is not clear which voxel material is the most optimal for all the different sky conditions. However, in June the simulations were not as accurate as in February, as the synthesized point cloud representation probably did not include a sufficient number of points and as the simulation system performed less accurately for days under clear sky conditions.
Consequently, the study demonstrates that voxel representations with sizes around 0.10 m provide the most reliable results for tree modeling in daylight simulations in February, while clarifying the limitations that gave rise to to the poorer performance of alternative representations and of June simulations.

Acknowledgments

I would like to express my gratitude to my academic supervisors, *Azarakhsh Rafiee* and *Eleonora Brembilla*, for their guidance and valuable feedback, as well as for their patience during some difficult days for me. The brainstorming sessions with them were always insightful and greatly contributed to the development of this research. I would also like to thank my co-reader, *Edward Verbree*, for his valuable feedback on my text, which helped me present my work and results more clearly to future readers.
Many thanks go to the Green Village staff, in particular *Marijn Leeuwenberg*, for providing the sensor data of the CCC building and kindly answering all my questions related to it.
Also, words cannot express my gratitude to my *family*, who gave me precious emotional support when times were difficult and I felt trapped and uncertain. Their encouragement gave me the strength to keep going and helped me regain faith in myself and my abilities.
Of course, a massive thanks to the friends I made here in the Netherlands, both within and outside the program. Even though some of them have left the country and I can no longer see them regularly, I still think of them often. They helped me unwind and disconnect when needed, bringing balance and joy throughout this journey.

Contents

List of Figures

List of Tables

Acronyms

BPS Building Performance Simulation . . . 1
LiDAR Light Detection and Ranging . . . 120
AHN Actueel Hoogtebestand Nederland . . . v
RD Rijksdriehoekscoördinaten . . . 25
NAP Normaal Amsterdams Peil . . . 25
CCC Co-Creation Centre . . . xiii
UAV Unmanned Aerial Vehicle . . . 2
TLS Terrestrial Laser Scanning . . . 3
ALS Airborne Laser Scanning . . . 3
LOD Level of Detail . . . 120
K-NNG K-Nearest Neighbors Graph . . . 14
DEM Digital Elevation Model . . . 13
MST Minimum Spanning Tree . . . 14
BRDF Bidirectional Reflectance Distribution Function . . . 19
DSM Digital Surface Model . . . 22
GPS Global Positioning System . . . 9
INS Inertial Navigation System . . . 7
DGPS Differential Global Positioning System . . . 7
IMU Inertial Measurement Unit . . . 9
TMY Typical Meteorological Year . . . 15
GHI Global Horizontal Irradiance . . . 15
DNI Direct Normal Irradiance . . . 15
DHI Diffuse Horizontal Irradiance . . . 15
CPU Central Processing Unit . . . 51
DBSCAN Density-Based Clustering . . . xii
EPW EnergyPlus Weather . . . 39
SoDa Solar Radiation Data . . . 32
CAMS Copernicus Atmosphere Monitoring Service . . . 15
RMSE Root Mean Square Error . . . xiii
MBE Mean Bias Error . . . xiii

1. Introduction

The light from the Sun is invaluable, since it fuels the entire ecosystem on Earth with energy. In this sense, natural light is widely acknowledged to greatly benefit people's health. Specifically, it supports the human circadian system, which regulates biological rhythms. Disruptions in these rhythms can lead to various health problems, including neurobehavioral disorders, cardiovascular problems, immune dysfunction, and hormonal imbalances (Hraska, 2015) [34]. In addition, mental health is also affected by natural light. People who are not exposed to adequate hours of daylight are prone to experience sadness, fatigue and clinical depression. Also, interaction with natural light reduces symptoms of Seasonal Affective Disorder (SAD), which is type of depression related to reduced daylight during winter period (Bertani et al., 2021) [11].
Urban designers recognize the importance of natural light and therefore integrate it into urban planning and architectural design. Beyond enhancing well-being, it mitigates the environmental footprint of buildings. Insufficient exposure to sunlight increases energy consumption for artificial lighting and heating (Bazir et al., 2024) [9]. The interest in energy efficiency was sparked in the advent of the 1980s oil embargo leading to the development of Building Performance Simulation (BPS), supported by advances in personal computers. Over the following decades, continuous improvements in computer hardware further enhanced BPS accuracy and usability and resulted in making BPS substantial for design decision making (Ayoub, 2019) [6].
In addition, trees play a vital role in the urban environment by shaping microclimates. Their shade and cooling effect contribute to lower temperatures, improving thermal comfort for city residents. From this perspective, the inclusion of trees in urban daylight analysis is essential to accurately estimate daylight availability (Balakrishnan and Jakubiec, 2022) [7].
However, there are some challenges about representing accurately the trees so as to be used in a daylight simulation. Characteristics such as height, branching structure, leaf density, and crown permeability are often disregarded by using generic 3D models to represent trees. Furthermore, seasonal variation adds further complexity, as many species undergo substantial changes dramatically throughout the year. Consequently, the condition of trees should be considered in relation to the specific time period of the daylight simulation.
To address this complexity, point cloud data can be employed not only to represent trees, but also as a basis for generating a variety of other tree 3D representations. In general, 3D representations can range from simplified to highly detailed, reflecting low or high idealization precision, which expresses how accurately real-world objects are defined and represented after measurement. In the context of trees represented with point cloud data contribution, the degree to which a 3D representation aligns with the actual tree structure depends on the quality of the point cloud data and, consequently, on the chosen representation method.
For instance, the tree crown, which is widely regarded in studies on urban microclimate (Xu et al., 2021) [73] can be accurately described by point cloud data or reconstructed using skeletonization algorithms that fit cylinders to branch points, provided the point density and distribution are sufficient and noise is limited (Du at al., 2019) [22]. On the other hand, when point cloud data are sparse, tree crown representations based on geometric abstractions, such as shapes derived from tree tops, crown width, crown base (de Groot, 2019) [32] or convex hull and alpha shape of branch points ((Xu et al. (2021) [73], de Groot (2019) [32], Zhu et al. (2008) [20]), can be adequate in many cases. However, they may fail to capture the actual canopy structure due to limited point density and are not suitable for studies requiring leaf-off tree representations or analysis of in-crown structure. Lastly, although voxel-based representations derived from point cloud data can compensate for inadequate point density through appropriate aggregation, which is dependent on voxel spatial distribution and size (Buchalová et al., 2025 [13]), and capture canopy porosity by integrating point counts or

laser return intensity, the resulting representation deviates from the actual canopy structure (Xu et al., 2021 [73]).
Regarding trunk representations, beyond direct point cloud data, many studies focus on fitting cylinders derived from skeletonization applied to point cloud input (Du et al. (2019) [22], Fan and Nan (2020) [27]). The scanning method plays a significant role in trunk reconstruction. Point clouds may be obtained from either in-situ laser scanning by Unmanned Aerial Vehicle (UAV) or Mobile Laser Scanner or from airborne Light Detection and Ranging (LiDAR) surveys. Airborne laser scanning often provides an insufficient number of points representing tree trunks compared to other laser scanning methods, due to the obstruction by the crown or unfavorable viewing geometry from the aircraft to the tree and the inability to intervene in the scanning process.
Lastly, several studies have already addressed the representation of leaves. Beyond direct point cloud representation, leaves can also be reconstructed using approaches that produce realistic surfaces. These approaches involve either model-free methods that rely solely on the given point cloud data, or model-based methods that incorporate additional factors such as predefined parametric surfaces or physiological characteristics of leaves (Ando et al., 2021 [4]). Once again, point cloud quality plays a critical role in all of these methods, with its influence being particularly strong for the first-mentioned representations. Alternatively, leaves can be represented by predefined surface models that are entirely independent of the input point cloud (Du et al., 2019 [22]). However, especially in urban-scale studies, leaves are often generally not individually represented, but rather leaves are considered only in an aggregated form, influencing canopy volume, density, and light transmittance (Xu et al., 2021) [73].
Studies of this nature can be facilitated by the availability of open-source geospatial databases. In the Netherlands, the AHN program provides open georeferenced point cloud data in the compound CRS Amersfoort / RD New + NAP height (EPSG:7415), using Amersfoort / RD New (EPSG:28992) for horizontal coordinates and NAP height (EPSG:5709) for elevations. The program began in 1997 with AHN 1 and was completed in 2004, continued with AHN 2 between 2007 and 2012, offering improved accuracy, and proceeded with AHN 3 starting in 2014, which introduced open data access for AHN 1, 2, and 3 and was completed in 2019. Next, AHN 4 was implemented between 2020 and 2022. Lastly, AHN 5 acquisition covers years from 2023 to 2025.
For this research, the study area was the Co-Creation Centre (CCC) building in the Green Village of TU Delft Campus and nearby area with Acer platanoeides trees. Using AHN 5 point cloud data, several tree 3D representations were generated. Then, daylight simulations were performed in Rhino with the ClimateStudio plugin, which is widely adopted among researchers and practitioners, using illuminance measurements from sensors on the west facade of the CCC building as reference. These simulations were used to identify the tree 3D representation which led to the best simulated illuminance values.

1.1. Research motivation

The motivation for this research stems from two main factors. First, most urban daylight simulation studies that incorporate trees rely on hemispherical imagery or point cloud data obtained from Terrestrial Laser Scanning (TLS) or UAV photogrammetry/ LiDAR rather than Airborne Laser Scanning (ALS). Consequently, there is a need to explore more which type of tree 3D representation derived from ALS point cloud data can provide reliable results in daylight simulation, since these data are widely used by both researchers and practitioners.
Additionally, despite growing attention to trees in urban environmental modeling, seasonal variations in tree conditions are often overlooked. As a result, most studies do not distinguish between leaf-on and leaf-off stages in daylight simulations, treating trees as static shading elements throughout the year. This simplification can lead to significant inaccuracies, as the presence or absence of foliage greatly influences light transmission, shading patterns, and overall daylight availability within urban environments. This research therefore aims to address this limitation by explicitly accounting for seasonal tree variation in daylight simulation analysis.

1.2. Research Objectives

This section presents the main research question and sub-questions that will be answered through this MSc thesis. The primary research question is:

To what extent can the tree ALS point cloud data increase the accuracy of daylight simulations?

The sub questions that emerge are:

1. **How can ALS point cloud data be used for tree 3D representation in urban daylight simulation?**
2. **What is the difference in accuracy of the results of the daylight simulation between diverse tree 3D representation approaches?**
3. **How can the results of daylight simulation be evaluated?**
4. **What is the impact of seasonal alterations of the tree canopy in urban daylight simulation?**

1.3. Scope of Research

This research aims to investigate how a tree ALS point cloud can be leveraged in daylight simulation within an urban environment and examine the results in terms of their accuracy and alignment with the real-world conditions.
For this research, the study area was the CCC building in the Green Village of TU Delft Campus, which is a sustainable glass building, where a diversity of research projects is supported (**Figure 1.1**) and the nearby tree area.

Figure 1.1.: CCC Building and nearby tree area (source: Co-Creation Centre and Nonohouse [43])

On the roof of CCC Building four illuminance sensors are installed namely global horizontal sensor, roof east sensor, roof west sensor and roof south sensor. Theodoropoulou (2023) [60] provided the insightful scheme shown at **Figure 1.2** depicting the positions of the sensors. These sensors measure illuminance at their respective positions every five minutes, which is recorded in UTC time.

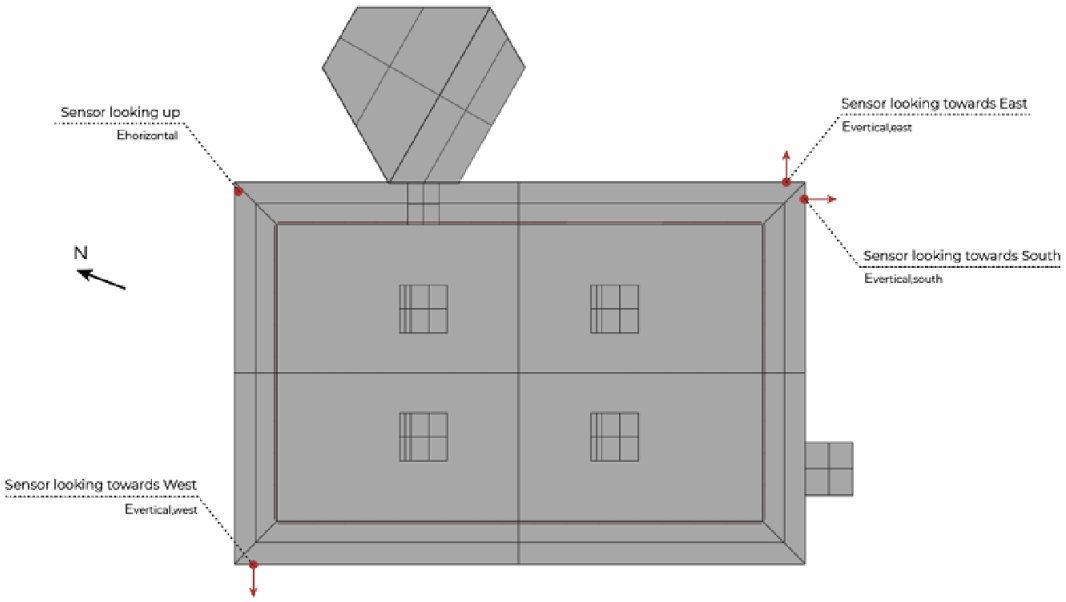


Figure 1.2.: Scheme of the distribution of the sensors on the roof of CCC building (Theodoropoulou, 2022)

Information on tree species in the study area was first sought in the dataset Bomen in beheer door gemeente Delft [30], but no records were available. Field visits were therefore conducted, and three mobile applications—Plantify [18], PlantNet [51], and PictureThis [50]—were used for identification. All three identified the species as Acer platanoides (Norway maple).
The forestry area sizes about 3,900 m² and consists of approximately 50 Acer platanoides. It is a medium to large deciduous tree that can grow to a mature height of up to 40 meters, depending on the conditions in which it is cultivated (Invasive Species Spotlight: Norway Maple [12]). Leaf drop begins in autumn, when day lasts approximately 10 hours. Before the leaves, the flowers appear in spring for two to three weeks. Then, Acer platanoides seeds need just three months of exposure to temperatures below 4 °C to germinate and typically sprout in early spring. Lastly, its leafout generally begins, when air temperatures reach approximately 12°C and daylight extends to at least 13 hours (Acer platanoides [70]).

The geospatial data employed in this study were obtained from the AHN 5 point cloud (LAZ format), covering the study area. To examine seasonal foliage variations, a synthetic point cloud was generated, consisting of:

1. Points from cleaned AHN 5 point cloud containing only the trees of the study area.
2. Points from samples on synthetic leaves created using the AdTree algorithm [22].

The geometrical representations which have been constructed for the trees using each of the point cloud files are demonstrated at **Table 1.1** :

Table 1.1.: Geometrical Representations

Point Cloud - based Case
Alpha Shape Case
Voxel Grid Case
Convex Hull Case

Next, the ClimateStudio plugin within Rhino was used to perform two types of daylight simulations. The first simulation type employed the representations based on the cleaned AHN 5 point cloud, corresponding to February conditions, while the second one utilized the synthesized point cloud, representing June conditions. Lastly, the simulated illuminance values were analyzed based on the records of the west sensor for February and June 2023.

1.4. Thesis Outline

Chapter 2 reviews related work, covering point cloud acquisition, tree 3D representation methods, and daylight simulation principles. Chapter 3 describes the research methodology, including the overall pipeline and evaluation strategy. Chapter 4 presents the technical implementation, detailing the data, tools, and processing workflows. Chapter 5 provides the analysis and evaluation of the simulation results. Finally, chapter 6 concludes with the main findings, discusses limitations, and suggests directions for future work.

2. Theoretical Background and Related Work

This section reviews past research relevant to the topic of this thesis. The collected information provided knowledge, inspiration and valuable information for decisions regarding the methodology and approaches to addressing the challenges in this research. This section will focus on three main components: the data collection methods previously used for tree point cloud, a discussion of common approaches to tree 3D representation, the background of daylight simulation and past studies on methods to estimate impact of trees in urban daylight simulations.

2.1. Data collection methods

The first step for tree 3D representation is the data collection. According to Okura (2022) [49] one of the typical approaches for this step is implemented via 3D laser scanners, which measure the distance traveled by emitted light using phase differences. The output of this method is a point cloud.
3D laser scanners appear in different forms. Du et al. (2019) [22] and de Groot (2020) [32] utilized the AHN 3 point cloud data from the open-source airborne (LiDAR) datasets of the Netherlands for their research on automatic tree 3D reconstructions. Regarding the way the airborne LiDAR system functions, it estimates the range between the airborne sensor and a ground target by measuring the time of flight between emission of a laser pulse and detection of the backscattered return from that target. At **Figure 2.1** from Dowman (2004) [21], the combined airborne LiDAR system can be illustrated. Except for the laser scanner, the aircraft is equipped with a Global Positioning System (GPS) receiver and Inertial Navigation System (INS). In order to determine an accurate global position, Differential Global Positioning System (DGPS) is implemented. It is based on the fact that any two receivers that in relatively close proximity will be affected by similar atmospheric errors. It involves two GPS receivers, the one is at accurately predefined position (base or reference station) and the other on the aircraft. The base station receiver determines its position from satellite signals and computes the offset between its known true position and the GPS-derived position, which is affected by atmospheric delays, satellite clock offsets, and orbit errors. This offset is used to generate a correction that is transmitted to users. The aircraft then applies the correction from the nearest base station to enhance its own GPS solution ("Computational modelling of terrains" notes [38], [26],[25]). At the same time, orientation is obtained from the INS, which estimates yaw, pitch, and roll using measurements from the onboard Inertial Measurement Unit (IMU). ("Computational modelling of terrains" notes [38])

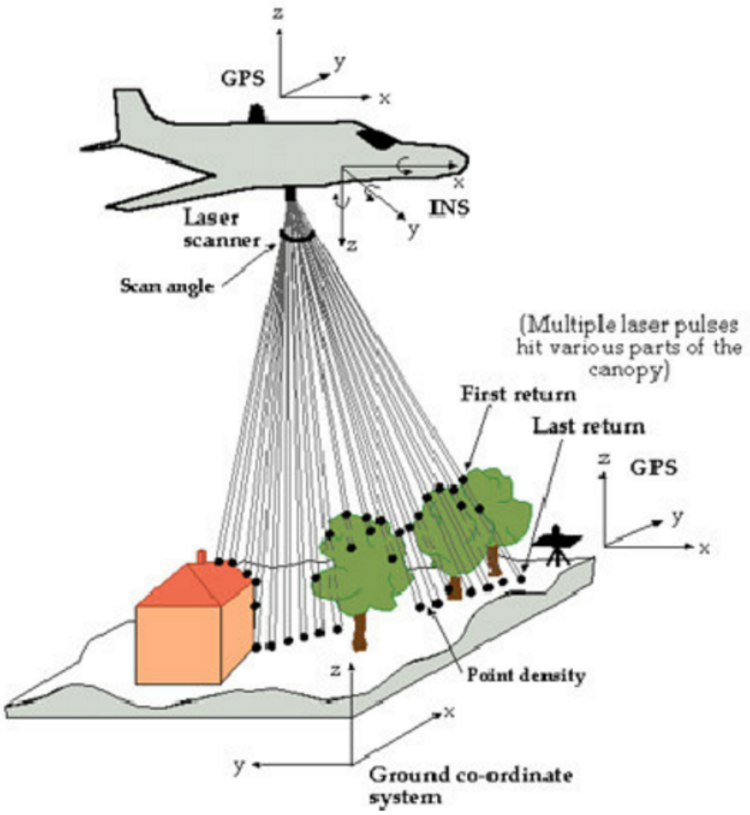


Figure 2.1.: Combined airborne LiDAR system (Dowman, 2004)

A key characteristic of airborne LiDAR tree measurements is the ability to record multiple returns from one pulse, first from the top of the canopy, then from lower layers and potentially the ground if the beam passes through gaps, whenever the backscattered signal exceeds detection thresholds (Chazette et al., 2016)[14].

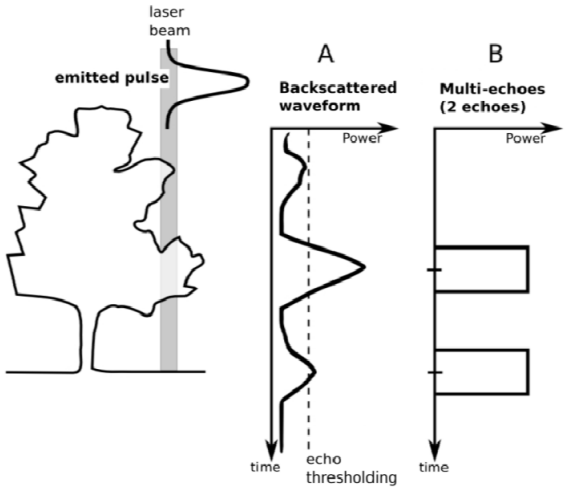


Figure 2.2.: Pulse traversing the canopy yields a first return or echo from a high branch, then later echoes from lower layer (Chazette et al., 2016).

Although airborne LiDAR is highly useful, sources of error from acquisition to final product should not be overlooked. First, at the acquisition stage, LiDAR sensor properties (e.g. pulse repetition frequency, scan angle, sensing range, and scan time) can cause significant artefacts in the point cloud. In addition, environmental conditions (weather and atmospheric effects) and navigation errors from the GPS receivers and the onboard IMU, together with limits in positioning and timing resolution, can also reduce accuracy ("Computational modelling of terrains" notes [38], (Kharroubi et al., 2022) [37]). Moreover, occlusion is a major source of error in airborne LiDAR measurements. More specifically, the relative geometry between the aircraft and ground objects can produce variable flight-line overlap and occlusion, leading to spatial inconsistencies in point density ("Computational modelling of terrains" notes [38]).
For forested environments, some researches have been conducted to predict the effects of occlusion caused by airborne LiDAR scanning for applications such as generating digital elevation models (DEMs) and analyzing forest structure. In such researches, laser scanning can be implemented through UAV, which provides researchers and surveyors with greater control over the data acquisition process. Gassilloud et al. (2025) [28] and Maio et al. (2025) [44] using (UAV), sensor position reconstruction and a ray tracing algorithm to estimate occlusion, provided evidence about the effect of flight UAV parameters on the level of occlusion and made some recommendations for survey planning.

2.2. Approaches tree 3D representations

Trees can be represented geometrically in multiple ways. According to Okura (2022) [49], relevant approaches include point cloud, voxel, mesh, parametric surface, and primitive-based representations.
Point cloud representation is widely used, because it provides an explicit and detailed depiction of trees, as with other 3D objects. **Figure 2.3** illustrates an example from the study point cloud in this thesis. The scene captures a variety of objects, ranging from streetlights to tall buildings, as recorded by airborne LiDAR.

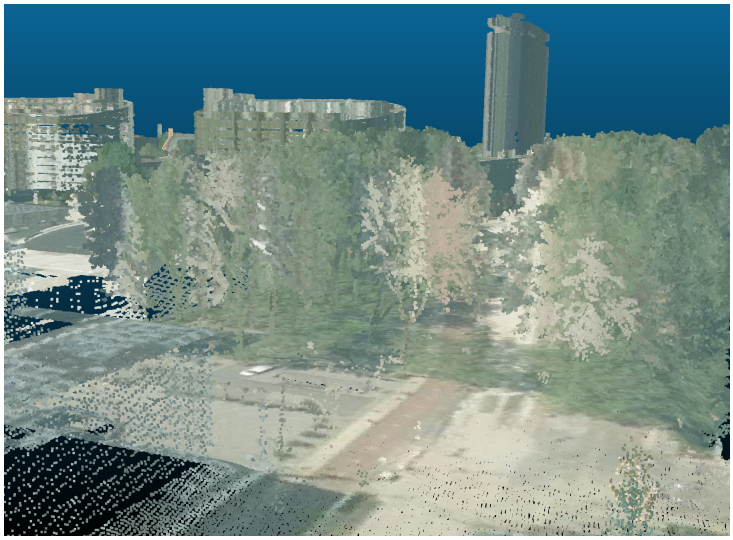


Figure 2.3.: Example from the AHN 5 dataset (tile 37EN2_16.LAZ), illustrating various objects captured by airborne LiDAR.

About voxel representation, voxel is a distinctive object in 3D grid, which is used to describe a 3D object or field. The typical voxel model consists of a standard regular 3D grid with box shaped voxels of the same size along every axis functioning as a 3D equivalent of a 2D pixel. (Wikipedia: Voxel [72]) A set of voxels, including a highlighted one is depicted at **Figure 2.4**. However, variations in voxel geometry exist, such as grid of voxels of different sizes along one or more axes, or skewed voxel orientations, or alternative shapes such as octahedra ("3D Modelling of the Built Environment" notes [5]).

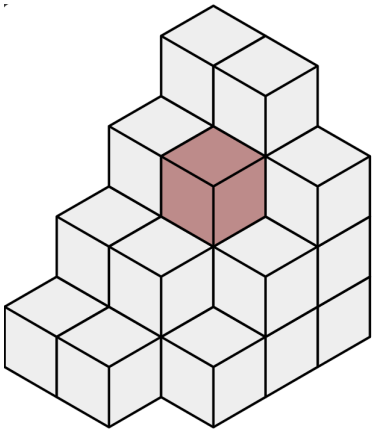


Figure 2.4.: Example of a voxel grid, with a single voxel in the shade. (Source: Vossman and M. W. Toews [66] (CC BY-SA 2.5)).

This representation can support the visualization and analysis of diverse scientific data, including geospatial datasets. For example, Li et al. (2024) [39] introduced SVDTree, a hybrid tree modeling approach that applies a denoising diffusion model to a masked tree image, transforming it into a Semantic Voxel Structure (SVS) where voxels encode geometric and semantic categories (trunk, branches, leaves), which then guide 3D tree reconstruction through a skeletonization method and a space colonization algorithm (**Figure 2.5**).

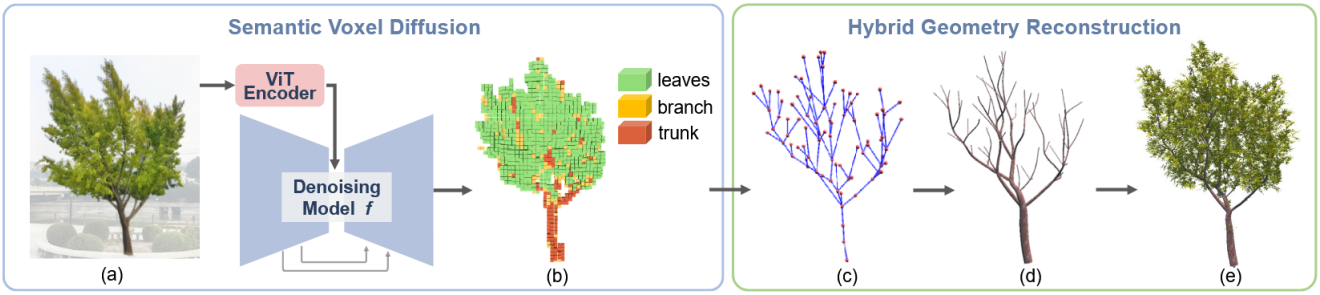


Figure 2.5.: SVDTree processing chain (Li et al., 2024)

Additionally, voxelization can be applied to convert point cloud data into a voxel grid, where each voxel stores geometric and attribute information derived from the points it contains. The process begins by defining the bounding box of the point cloud and determining the number of voxels that fit along each axis according to the chosen voxel size. Based on these calculations, the point cloud space is segmented into discrete voxels, each encoded with an object identifier and associated value(s) representing the characteristics of the points within it (Xu et al. (2021) [74], "Voxelization Algorithms for Geospatial Applications: Computational Methods for Voxelating Spatial Datasets of 3D City Models Containing 3D Surface, Curve and Point Data Models" notes [48], "3D Modelling of the Built Environment" notes [5]). The procedure can be modified by using sparse voxel models, which store only the occupied by points voxels instead of the entire domain defining the bounding box ("3D Modelling of the Built Environment" notes [5]).
Additionally, a dense spatial discretization of the regular computational grid is often required in regions where the objects of interest are located. The most common approach is to subdivide the coarse grid cells in these areas into smaller cells along each axis. However, if the designated area is not rectangular, this process may generate redundant fine cells outside the target region, leading to additional computational cost (Szucki et al., 2012) [57] (**Figure 2.6**).

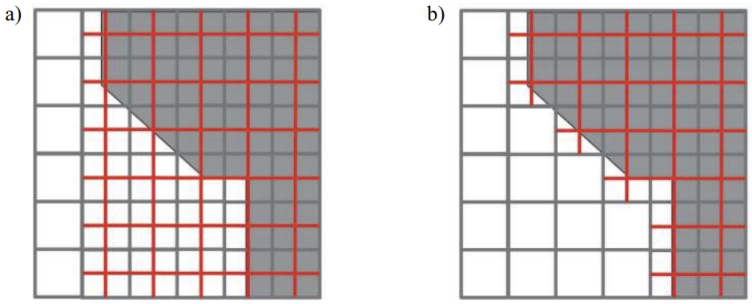


Figure 2.6.: Representation of selected resolution refinement: (a) Grid refinement of the bounding area defined by the object of interest; (b) Refinement of only the cells occupied by the object of interest, resulting in the creation of some unnecessary fine cells (Szucki et al., 2012).

In their research Weiser et al. (2021) [69] reconstructed segmented individual trees from terrestrial laser scanning (TLS) point clouds with stems as meshes while voxelizing crowns using different strategies (**Figure 2.7**).

Figure 2.7.: Individual tree point cloud and two of its different representation (Weiser et al., 2021)

In addition, mesh representations employ interconnected polygons (typically triangles) to form surfaces. Point/voxel-based representations are often transformed to polygon meshes. Two characteristic examples of mesh representations are convex hull and alpha shape. "Convex hull of a point set in Euclidean d-space is defined to be the smallest convex set containing the points" (Bentley et al., 1982) [10]. Given the aim of this research, only the 3D convex hull is of relevance (**Figure 2.8**).

Figure 2.8.: Example of convex hull of 3D points (Lorie Toma's lecture slides [63]).

On the other hand, the alpha shape can be regarded as a generalization of the convex hull. Intuitively, given a finite set of 3D points and α, a real parameter and assuming that the 3D space is filled with Styrofoam and the points are of a hard material, a sphere of α as radius acts as a carving tool and stops where the points are located. The obtained result, after straightening the edges, is the alpha shape. About parameter α, it controls the level of detail, enabling the representation of concave boundaries within a point set. If $\alpha = \infty$, the α-shape coincides with the convex hull of S. As α decreases, the α-shape contracts and progressively forms cavities. (Edelsbrunner and Mücke, 1994) [24] (**Figure 2.9**).

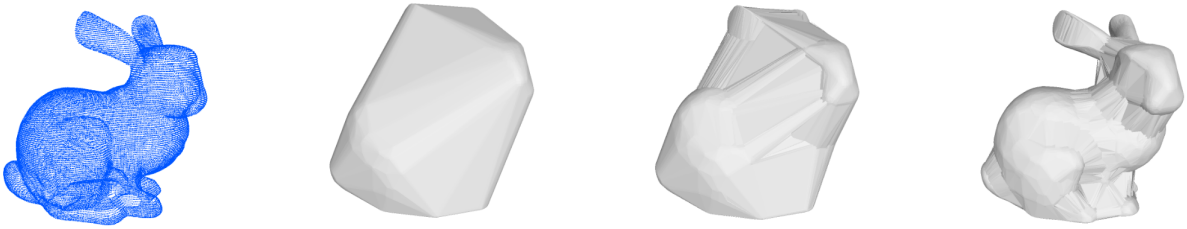


Figure 2.9.: Example of alpha shape implementations. From left to right: 1) original point cloud, 2) The convex hull of the original point cloud, 3) Aplha shape with α equal to 0.005 m, 4) Aplha shape with α equal to 0.001 m (Cao et. al, 2024)

Even though Weiser et al. (2021) [69] utilized mesh to reconstruct the stem of the trees, Zhu et al. (2008) [20] proposed using alpha shapes to reconstruct the outer boundary (silhouette) of tree crowns (**Figure 2.10**). It is also worth noting that although Trochta et al. (2017) [65] do not specifically study tree representations, their work underscores the importance of mesh-based tree representation. In particular, they developed the 3D Forest software, which extracts tree and crown parameters from TLS data, including crown surfaces and volumes computed using convex and concave hulls.

Figure 2.10.: Example of implementation of Zhu et al. (2008). From left to right: 1) original point cloud, 2) outer boundary points, 3) resulting boundary mesh model.

Moreover, specifically leaves can be represented following different approaches such as implementing methodologies that reconstruct 3D surfaces using the input point cloud, such as Poisson surface reconstruction, or fitting parametric surfaces to the point data, such as Non-Uniform Rational B-Splines (NURBS). In addition, Ando et al. (2021) [4] proposed a method that reconstructs a leaf using its skeleton extracted by the leaf point cloud, flat leaf shape produced by its skeleton, and a distortion component that captures 3D deformations deviating from the flat shape but constrained by the leaf skeleton.
de Groot (2020) [32] contributed valuable insights into constructing 3D tree model in various Levels of Detail (LODs). The first step was filtering vegetation from non-vegetation points based on standard deviation in height within neighborhoods (high values portrait vegetation points) and height offset of 2 meters and above. Next, Watershed Segmentation methodology was applied to segment individual trees from classified vegetation, using Digital Elevation Model (DEM) as input. Lastly, after cleaning the processed data, tree models used for different LODs were proposed depicted in **Figure 2.11**. For this research, LOD3.0 and LOD3.1 are relevant, since they focus on constructing the trunk implicitly using cylinders, while explicitly modeling the crown from point cloud data leading to more detailed tree 3D model compared to the other LODs.
It is worth-mentioning that, apart from Watershed Segmentation methodology, the last years a plethora of methods which execute tree segmentation from point cloud data. Two categories of these techniques are bottom-up and top-down methods. The bottom-up methods initially perform point clustering of trunk

points and subsequently connect segments from it into individual trees (Jiang et al., 2023) [35], while the top-down methods utilize points of tree peaks as seed points to identify separate trees (Zeng et al., 2025) [75]. Wang developed the *TreeSeparation* [68], which perform either both top-down or bottom-up tree segmentation on point cloud data after configuring the following parameters: `Searching radius`, `Vertical resolution`, `Minimum number of points per cluster`. Another approach on tree segmentation is executed by making use of graph pathing algorithms. Wang et al. (2021) [67] proposed a combined graph structure that integrates a k-nearest neighbors graph (K-NNG) for k = 10 and Delaunay triangulation discarded the long edges of Delaunay graph and increase connectivity between the points that the same time. Using the hybrid network, nodes are grouped by tracing paths to their lowest neighboring node. More specifically, all nodes that reach to the same lowest node are assigned to the same group.

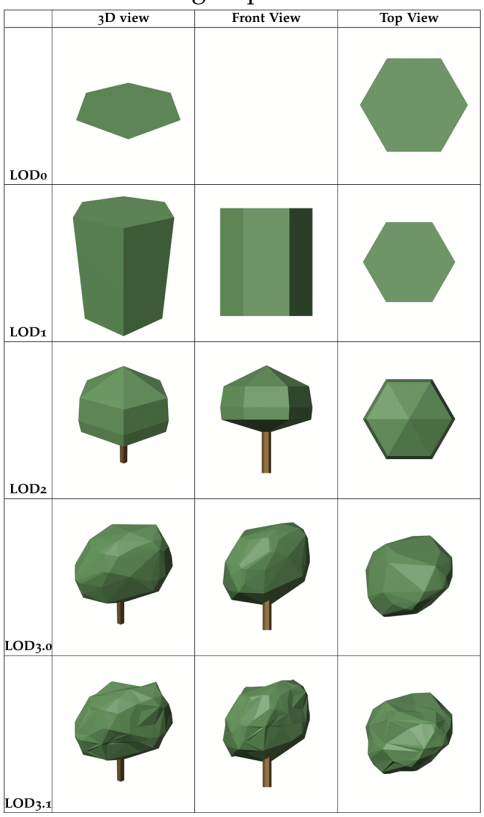


Figure 2.11.: Prosposed LODs by de Groot (2020)

Another common approach to 3D tree representation focuses on the tree's structural representation, as the form of branches is a fundamental characteristic of trees. Many implementations are graph-based, using point cloud as input and following skeletonization methods. These methods rely on connecting neighboring points and constructing branch structures by solving the shortest path problem (Okura, 2022) [49].
A notable example of skeletonization is the method developed by Du et al. (2019) [22], who generated an automated algorithm, namely AdTree, to implement tree 3D reconstruction using an individual tree point cloud. This method begins by extracting the initial tree skeleton applying Minimum Spanning Tree (MST) algorithm. The skeleton is then simplified by merging vertices, which were located quite close to each other, cylinders are fitted to approximate the geometry of the branches and trunk, followed by the addition of synthesized leaves to enhance realism.

2.3. Daylight Simulation

In the context of daylight simulation, the one of the most important metrics is illuminance. It describes the quantity of incident light that falls onto or illuminates a specific surface area. In SI units, illuminance is measured in lux (lx), which is equivalent to lumens per square meter ($\mathrm{lm} \cdot \mathrm{m}^{-2}$) (Illuminance [71]).
For daylighting it is necessary to involve modeling the sky's celestial hemisphere, which produces two location-based components: direct sunlight and diffuse daylight (Ayoub, 2019) [6] For this purpose, a specific sky model, which is a mathematical model of the real sky's luminance distribution or cumulative sky is applied to represent solar irradiance over a prolonged period (Murdoch, 1985) [46]. Ayoub (2019) [6] refers to the sky models by the chronological order they were developed. One of the most widely used sky models are the following:

1. **Uniform Sky:** It was based on a simplified luminance distribution and did not take any direct sunlight into account.
2. **CIE Standard Overcast Sky:** It is widely used for Daylight Factors in cloudy European climates and disregards direct sunlight due to prevailing cloudy conditions.
3. **CIE Standard Clear Sky:** It represents sunny conditions.
4. **Intermediate Sky:** Since both CIE Standard Skies depict two extreme conditions, Intermediate Sky satisfies the transition between clear and overcast skies and supports a more complex luminance distribution.
5. **Perez All-Weather Model:** It is popular among implementations of daylight simulation, predicts solar irradiance and is consistent with weather data files.
6. **Utah Sky:** It is a physically-based color gradient model and models and accounts for non-visual effects of daylight.

Subsequently, especially for the external luminance estimation, solar irradiance measurements are exploited, commonly acquired from weather stations along with other meteorological data (Crawley, 2007) [20]. These weather datasets provide typical single-year records representing historical data for particular locations. They include hourly records in UTC time of diverse weather elements, among which Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiance (DHI) are included and needed to estimate diffuse daylight. Such datasets are commonly available in the form of Typical Meteorological Year (Typical Meteorological Year (TMY)) files, which are widely used in daylight simulations (Ayoub, 2019) [6], primarily distributed in the EPW format.
Below the definitions of GHI, DNI and DHI by "User Guide to the CAMS Radiation Service (CRS)" are provided [19]:

- "Global Horizontal Irradiance is the radiation that is received by a horizontal plane from all directions." **[W/m^2]**
- "Direct Normal Irradiance is the part of the radiation that is received from the direction of the sun by a plane facing the sun." **[W/m^2]**
- "Diffuse Horizontal Irradiance is the part of the radiation that is received on a horizontal plane from all directions except that of the sun." **[W/m^2]**

For any location in Europe time series of these solar radiation data are provided by Copernicus Atmosphere Monitoring Service (CAMS). **Figure 2.12** shows the scattering and absorption of solar radiation by clouds, aerosols, ozone, and water vapour, resulting in the separation of irradiance into its main components, DNI, DHI, and GHI. The Heliosat-4 method estimates the downwelling shortwave irradiance reaching the Earth's

surface under all-sky conditions. It is based on two models, McClear and McCloud. The former receives aerosol properties, total column water vapour and ozone content as provided by the CAMS global forecast and reanalysis services every 3 hours and computes clear-sky irradiance. At the same time, the latter takes cloud properties from images of the Meteosat Second Generation (MSG) satellites in 15 min temporal resolution using the cloud processing scheme APOLLO (AVHRR Processing scheme Over cLouds, Land and Ocean) and estimates the attenuation of radiation caused by clouds. Eventually, after combining and processing the products of both models, this method provides global, direct, and diffuse irradiance on a horizontal plane, as well as direct irradiance on a plane perpendicular to the sun's rays ("User Guide to the CAMS Radiation Service (CRS)" [19], Schroedter et al. (2022) [53], Gu et al. (2017) [52]).

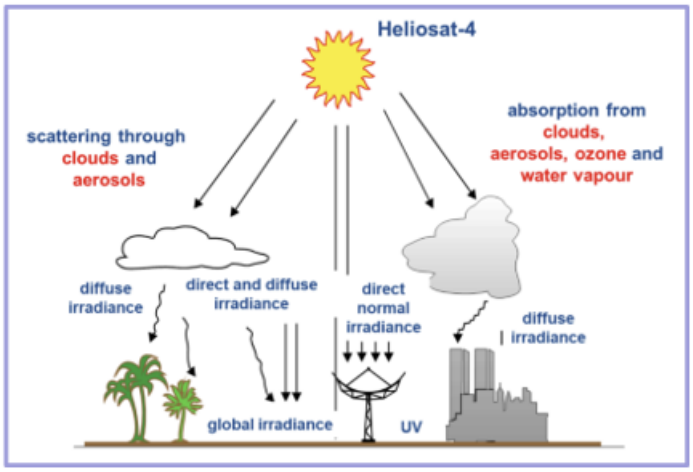


Figure 2.12.: Illustration of solar irradiance interaction with the atmosphere and derivation of Global Horizontal Irradiance, Direct Normal Irradiance, and Diffuse Horizontal Irradiance (source: User Guide to the CAMS Radiation Service (CRS) [19])

Before proceeding to the practical implementation, a few significant concepts should be addressed. Global illumination is a computer graphics technique that simulates rays from a light source which bounce off the surfaces of objects of a scene many times (indirect illumination), rather than only accounting for rays traveling directly from the light source (direct illumination). This approach produces more realistic and natural-looking virtual environments (**Figure 2.13**). In theory, phenomena such as reflections, refractions, and shadows are considered forms of global illumination, since their simulation requires accounting for how one object influences the lighting of another, and not only considering direct light from a source.

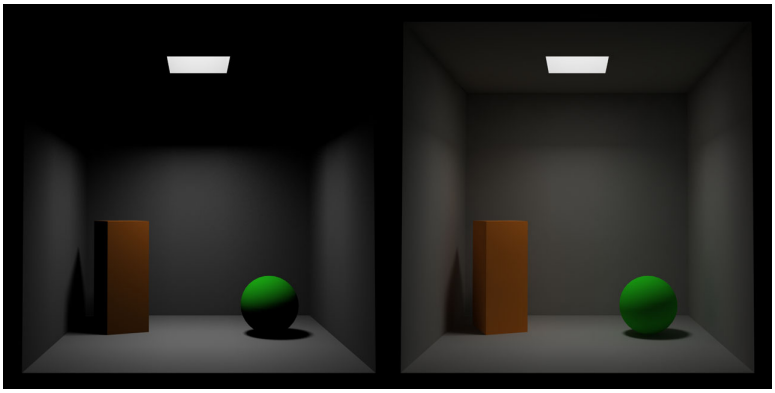


Figure 2.13.: Scene with direct illumination only (left) compared to a scene including global illumination effects (right). (source: HOOPS Luminate: Global Illumination[58])

To simulate global illumination at every point on every surface in the scene, the *Rendering Equation* must be solved. It defines the radiance leaving a surface at position x in direction v and is expressed as:

$$L_{\text{out}}(x, \boldsymbol{v}) = \int_{\Omega} L_{\text{in}}(\boldsymbol{\omega})\, f_r(x, \boldsymbol{\omega}, v) \cos\theta \, d\boldsymbol{\omega} + E_{\text{out}}(x, v) \tag{2.1}$$

where:

- $L_{\text{out}}(x, v)$ = outgoing radiance at point x in direction v (from point x toward the scene observer),
- $L_{\text{in}}(\boldsymbol{\omega})$ = incoming radiance from direction $\boldsymbol{\omega}$,
- $f_r(x, \boldsymbol{\omega}, v)$ = bidirectional reflectance distribution function (BRDF), which is the proportion of light reflected from direction $\boldsymbol{\omega}$ to $\boldsymbol{v}$ at position x,
- θ = angle between the incoming light direction $\boldsymbol{\omega}$ and the surface normal $\boldsymbol{n}$,
- $E_{\text{out}}(x, v)$ = emitted radiance at point x in direction $\boldsymbol{v}$,
- $\boldsymbol{\Omega}$ = the unit hemisphere centered on the surface normal $\boldsymbol{n}$ at point x, representing all possible incoming directions $\boldsymbol{\omega}$. (**Figure 2.14**)

The rationale of the Rendering Equation (2.1) is that the incoming radiance terms on its right-hand side generally correspond to the outgoing radiance from other points in the scene. This suggests that this mathematical problem has a recursive nature which introduces considerable complexity. However, the value of $f_r(x, \boldsymbol{\omega}, v)$ typically diminishes after several bounces, implying that later contributions to the integral are limited. This property allows the integral to be approximated, while still producing high-quality rendering results.

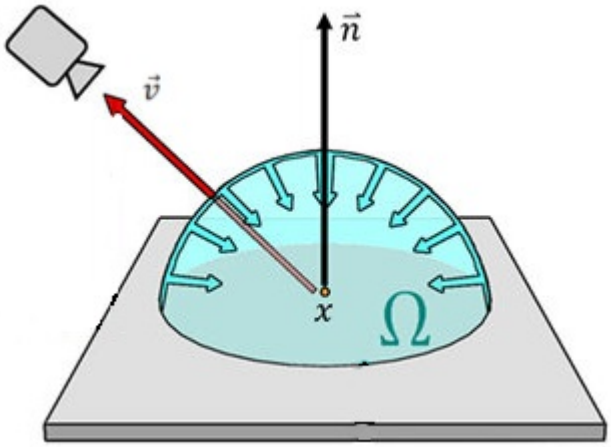


Figure 2.14.: Representation of outgoing radiance at point x in direction v (red arrow), resulting from the contribution of incoming radiance $L_i(x, \omega_i)$ from all directions ω_i within the hemisphere Ω centered on the surface normal n (Kahl, 2022 [36]).

Two algorithms used for scene rendering are *Ray Tracing* and *Path Tracing*. Firstly, the *Ray Tracing* algorithm is implemented by the following steps:

1. A primary ray is cast from the camera (observer's position) through each pixel into the scene.
2. The closest intersection of the ray with an object in the scene is computed.
3. At the intersection point:
    - A shadow ray is traced toward each light source to account for the direct illumination that reaches the point and is reflected toward the eye.
    - Reflection and refraction rays are generated according to the laws of optics, potentially intersecting other objects in the scene.
4. Step 3 is applied recursively for each new intersection, until a recursion limit is reached or further contributions become negligible (**Figure 2.15**).

When primary rays are traced from the viewpoint rather than from the light source, the process is referred to as *Backward Ray Tracing*. This approach is generally more efficient, as it considers only the rays that contribute directly to the image by reaching the viewpoint.

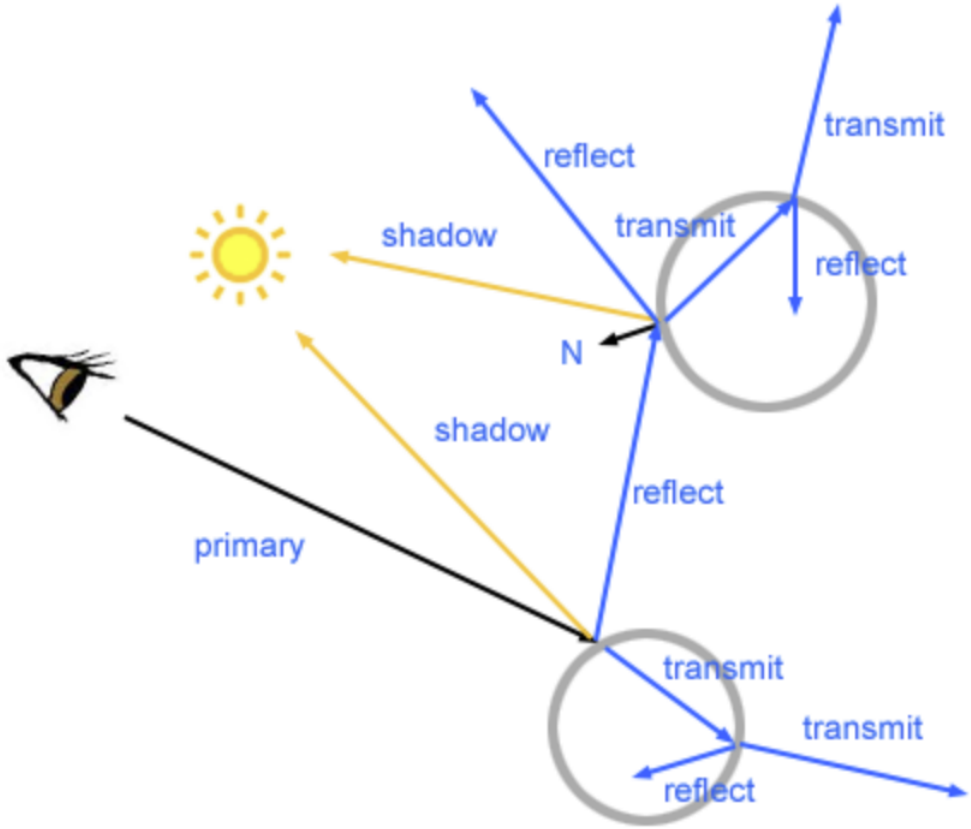


Figure 2.15.: Representation of recursion in the *Ray Tracing* algorithm. (source: Scratchapixel: Overview of the Ray-Tracing Rendering Technique [54])

With respect to the *Path Tracing* algorithm, it is a *Ray Tracing* technique that exploits the assumption that not all bounces contribute significantly to global illumination. It employs Monte Carlo integration, which is based on repetitive random sampling to compute numerical results. The steps that are followed according to this algorithm are the following:

1. Multiple primary rays are shot from the camera (observer's position) through each pixel into the scene.
2. The nearest point where the ray intersects an object in the scene is determined.
3. At the intersection point:
   - A shadow ray is cast toward each light source to determine the direct illumination, as in *Ray Tracing*.
   - A random direction is sampled (according to the surface's (BRDF) to trace a new ray.
4. Step 3 is applied recursively for each new intersection, while accumulating emitted light and reflected light contributions along this path, until a recursion limit is reached or further contributions become negligible.
5. In the end, all paths per pixel using Monte Carlo integration are averaged. (**Figure 2.16**).

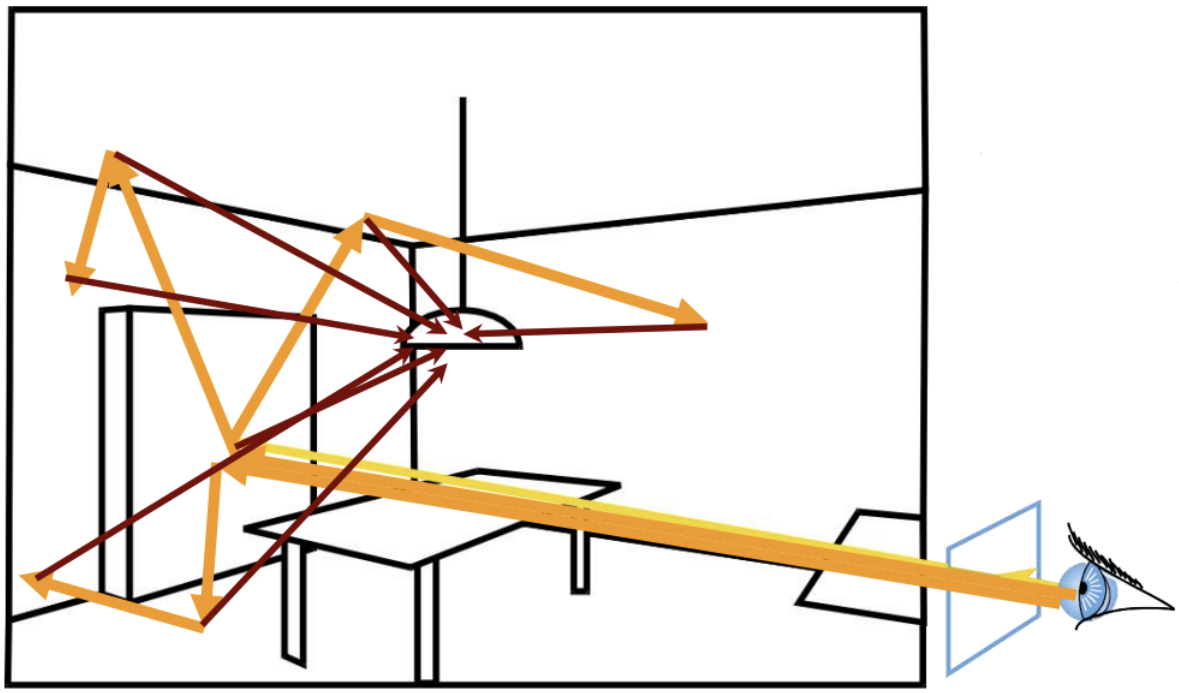


Figure 2.16.: Representation of ray interactions based on *Path Tracing* algorithm. (source: MIT OpenCourseWare: Computer Graphics [23])

A commonly used tool for performing daylight simulations is the ClimateStudio plugin for Rhino software. ClimateStudio models light behavior based on Radiance, a widely used, physically based industry standard engine (Climate Studio [17]). Regarding computation method ClimateStudio follows, it implements a stochastic (probabilistic) method, specifically progressive path tracing. This approach allows ClimateStudio to shoot a few rays from the sensor point(s) at a time or else per pass, instead of all possible ones and to collect starting estimates almost immediately and, then, there is a gradual denoising of the data as light paths are gathered (Climate Studio: Speed [17]). This, together with the GPU-based parallel computation, account for Climate Studio's high performance in terms of processing time without the results being less accurate (Climate Studio: Accuracy [16]).
Regarding annual simulation performance, ClimateStudio employs the Daylight Availability workflow. Even though detailed information about ClimateStudio's internal functioning is limited since it is not open source, it is understood that its implementation is based on a matrix-based approach combined with the Perez all-weather model to generate hourly results for various metrics, including illuminance. Within this framework, direct sunlight is calculated according to the solar position, while diffuse ambient light is derived from the 145-patch Tregenza sky subdivision. This method closely follows the Radiance 5-Phase method (Honningdalsnes et al., 2025) [33]. With regards to Radiance 5-Phase method, it builds upon and extends Radiance 3-Phase method, according to which the illuminance at a sensor point in an interior space is defined by the following equation (Tregenza and Waters (1983) [64]; Geisler-Moroder et al. (2017) [29]; Mie (2023) [45]):

$$\textbf{Illuminance}_{3-phase} = V \cdot T \cdot D \cdot S \tag{2.2}$$

where:

- V = view matrix correlating the sensor point with outgoing directions of the facade system,
- T = transmission matrix that reflects on the facade system,
- D = daylight matrix connecting incoming directions at the facade system with the sky patches,
- S = discretized sky luminance distribution as a vector or matrix based on the Perez all-weather model.

5-Phase method contributes to treating the direct solar component independently from the sky and inter-reflected solar components. According to it, the illuminance is calculated as follows (Geisler-Moroder et al. (2017) [29], Mie (2023) [45], McNeil (2013)[42]). Subramaniam (2017) [56] provides with a schematic representation of 5-Phase method, indicating how the aforementioned matrices contribute to the illuminance estimation, which is showed at **Figure 2.17**.

$$\textbf{Illuminance}_{5-phase} = \textbf{Illuminance}_{3-phase} - V_d \cdot T \cdot D_d \cdot S_d + C_{sd} \cdot S_{sun} \tag{2.3}$$

where:

- V_d = direct view matrix,
- T = transmission matrix that reflects on the facade system,
- D_d = direct daylight matrix,
- S_d = luminance vector or matrix derived only from the Sun,
- C_{sd} = direct Sun coefficient Matrix, which links the radiance from numerous sun positions to the direct illuminance measured at a sensor point.

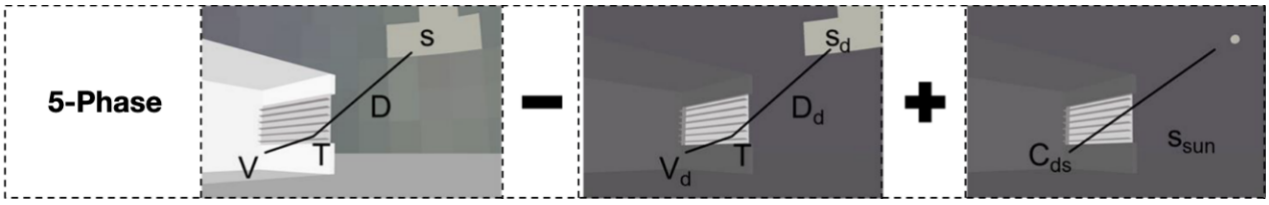


Figure 2.17.: Representation of 5-Phase method based on Equation (2.3) (Subramaniam, 2017)

In order to use the Daylight Availability workflow the following parameters are needed to be specified:

1. **Location:** The TMY file which corresponds to the location of the study area is specified here. Through the TMY file, hourly solar irradiance data required for environmental performance analysis can be obtained.
2. **Materials:** The materials of the objects, of which the scene for the lighting simulation consists, are assigned.
3. **Occupied Area(s):** The surface(s), on which the simulation will be performed, together with their sensor point(s) and surface normal vector(s), are specified.

2.4. Past researches on impact of trees on urban daylight simulations

The inclusion of trees in urban daylight simulations has been the focus of several studies over the years. With regard to tree reconstruction in urban microclimate simulations, the literature review conducted by Xu et al. (2021) indicates that point cloud data obtained from laser scanning or photogrammetry are widely adopted for this purpose, while ALS data collected from aircraft are used less frequently. With respect to this acquisition method, Lu et al. (2022) [41] proposed a methodology that simulates the shading patterns of the urban canopy derived from ALS data in the City of Vancouver, Canada. In this study, urban trees were represented as 3D models derived from ALS data collected in this city. Each tree model incorporated four main attributes: height, crown size, shape, and density. The first three attributes were obtained using an object-based image analysis (OBIA) approach applied to a Digital Surface Model (DSM) generated from the ALS point cloud data. For the fourth attribute, canopy density was calculated for each tree as the ratio of return points classified as vegetation in terms of height (higher than 3 m) to the total number of returns (both vegetation and ground). This density served as a proxy for light transmittance, quantifying the extent to which sunlight could penetrate each 3D tree model. The resulting models were then integrated into a Radiance daylight simulation, where each tree 3D model acted as a semi-transparent shading element, with its density controlling the degree of light blockage or transmission.
Addionally, Buchalová et al. (2025) [13] developed and compared methods for estimating solar radiation beneath tree canopies using LiDAR-derived point clouds in combination with GIS-based modeling. Two main approaches were implemented: the r.sun model within GRASS GIS, which uses rasters like DSM and raster to define clear sky solar radiation attenuation caused by the atmosphere and canopy and the Point Cloud Solar Radiation Tool (PCSRT), a voxel-based 3D model that builds regression surfaces within each voxel based on surrounding points and simulates the propagation of direct, diffuse, and reflected radiation for the specific period. The authors tested both methods in forested and urban environments using LiDAR data from different platforms, including terrestrial, unmanned aerial, and airborne laser scanning. For urban areas specifically, the study reports that TLS data achieved the highest point density, averaging around 3,991.41 points per square meter, while ALS data exhibited a much lower density, averaging 23.08 points per square meter. Their results showed that PCSRT provided higher precision in complex canopy conditions due to its detailed 3D representation. In addition, TLS offered high local accuracy but limited spatial coverage, and ALS data, despite its extensive coverage, exhibited reduced accuracy under dense canopy conditions due to lower point density. The study highlights the trade-offs between spatial extent and structural detail and concludes that voxel-based methods with high-density LiDAR are most effective for modeling canopy solar radiation in structurally complex environments. Another commonly adopted way to differentiate objects represented by voxels is to assign each voxel a value based on the number of points it contains or the intensity of the return laser beam (Xu et al., 2021) [73]. This information can be used to determine the radiative behavior of each voxel.
Lastly, Tian et al. (2023) [61] developed a hybrid modeling approach that integrates 3D point cloud data with surface-based methods to simulate how trees partially shade urban environments. A Deep Graph Convolutional Neural Network (DGCNN) was first applied to segment tree points from urban LiDAR point clouds, effectively distinguishing vegetation from buildings and ground surfaces. To address the limitations of existing irradiance models, the authors combined two matrix-based methods: Radiance, which accurately handles reflections and diffuse light from solid surfaces but oversimplifies trees, and Pyrano, which is a Python package for estimating solar irradiance on outside building surfaces and was used to model tree canopy transmittance based on point density, but lacks reflection modeling capability. By merging the two, they employed Radiance for built surfaces and Pyrano, utilizing DSM point clouds, for tree canopies, preserving the geometry and porosity of tree crowns.
The workflow involved computing a Daylight Matrix with Radiance, excluding trees, to capture surface reflections, generating a Tree Shading Ratio (TSR) matrix with Pyrano based on canopy point density, combining both with a Sky Radiance matrix derived from weather data and finally calculating the total irradiance at

sensor points. It was concluded that this hybrid framework allowed a realistic and computationally efficient simulation of partial shading effects from trees in complex urban settings, after assessing its alignment with the results of measurements with pyranometer in the study area.
Apart from point cloud data in urban daylight simulation, it is worth mentioning that many studies such as Balakrishnan and Jakubiec (2022) [7] capture sky-occlusion patterns of real trees affecting daylight availability of buildings in their study area in hemisphere images captured from diverse angles. Specifically, after processing hemisphere images, to capture the porosity of the trees from all directions around the horizon, they treated the tree canopy as a complex fenestration system, exploring different ways to represent it as a semi-transparent, volumetric filter composed of leaves and branches that interact with light through transmission, reflection, and scattering, rather than as a uniform shading object.

3. Methodology

This section outlines the methodology employed in this thesis. It consists of the description of the pipeline of this research, the connection of methodology and theoretical background provided in Chapter 2 together with some additional theoretical concepts and the explanation of the metrics used to evaluate the results.

3.1. Research Workflow

As mentioned in Chapter 1, in this research the extent of how much a 3D tree representation derived from AHN 5 point cloud data can enhance the accuracy of daylight simulation taking the seasonal variations into account. The steps of this research consist of the processing on the data, the creation of the geometric representations, the procedure leading to the daylight simulation and the evaluation of the simulation results. The research workflow of the research is illustrated at **Figure 3.10**.

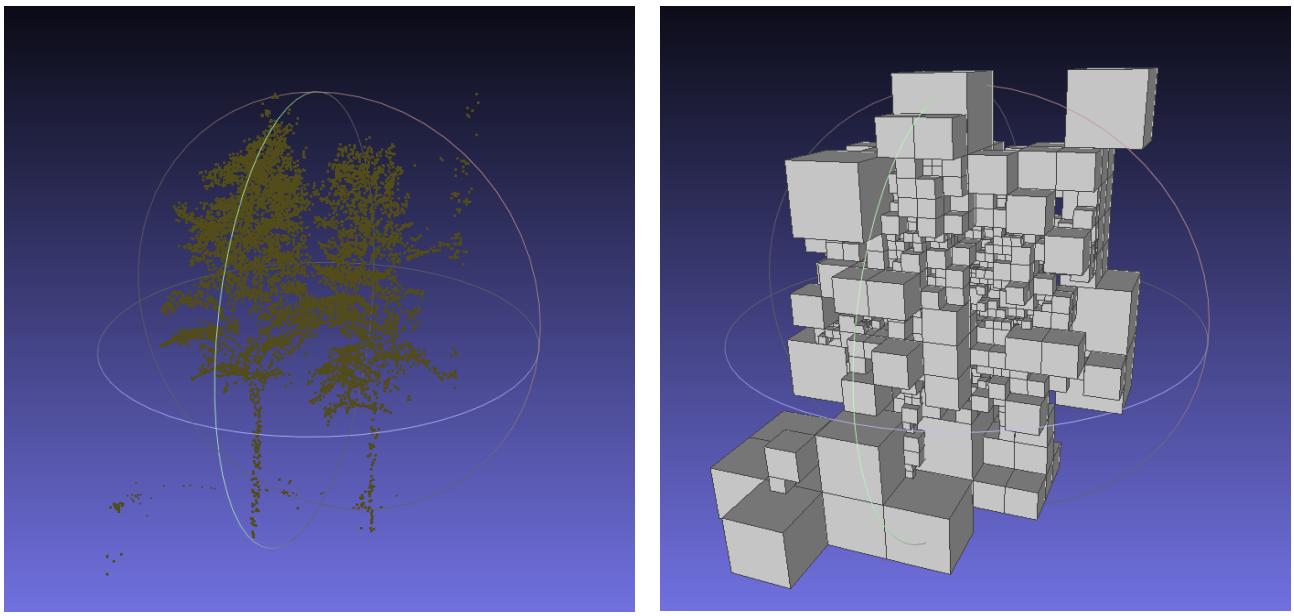
3.1.1. Data Processing

The first step is the adjustment of the AHN 5 point cloud data, so that only trees are isolated from the rest of the point cloud and studied. Next, the generation of the synthesized point cloud is realized via the contribution of AdTree algorithm [22]. Finally, a detailed 3D model of the CCC Building provided by the Green Village had to be georeferenced to Amersfoort / RD New + NAP to synthesize the scene with the 3D tree representation in the same coordinate system. This representation of the CCC Building was chosen for this representation, because it already contains a variety of surfaces representing different building components, such as glass facades, solid walls, the ground surface, and surrounding paths, which can later be linked to specific materials. This allows for a more accurate treatment of radiative transfer in the scene during simulation.

Geometric tree 3D representations

The second step involved the creation of aforementioned four 3D tree representations. The first case was the Point Cloud - based Case. Since the point cloud entity could not be detected in Rhino software, the points were converted into cubes with an edge length of 0.02 m, as this was the smallest size that could be represented in the software.
The second case was the Voxel Grid Case, inspired by Li et al. (2024) [39] and Weiser et al. (2021) [69], according to which the point cloud data were turned into voxel grid of a given voxel size. The chosen form of the voxel grid for this task is standard regular with cubic voxels of sizes of 0.03, 0.05, 0.07, 0.10, 0.30 and 0.50 m. Smaller voxel sizes were tested compared to this used in the voxel grid derived from ALS data in the study by Buchalová et al. (2025) [13], in order to assess how a finer level of spatial detail affects the results of the daylight simulations.
In addition, a corresponding resolution refinement method described by Szucki et al. (2012) [57] to point

cloud, which would improve voxel resolution in regions with higher point density, was not implemented. Applying such an approach would significantly increase the number of voxels, since coarse voxels with high point density would be subdivided, leading not only to longer computational time but also to a higher number of surfaces for light rays to interact with. Also, unnecessary fine voxels along the boundaries of the tree area would be generated. The combination of these additional refined surfaces and those from the unpartitioned coarse voxels would result in premature light attenuation and, consequently, an excessive underestimation of the simulated illuminance values (**Figure 3.1**).



(a) Example of input point cloud.

(b) Output of voxel grid with varying voxel size.

Figure 3.1.: Example of the voxelization process applied to a tree point cloud, illustrating the transformation from the original point cloud (a) to the corresponding voxel grid representation with varying voxel sizes (b).

Regarding voxel orientation, it was decided to use axis-aligned voxels based on the coordinate system of the point cloud data. By definition, the first step of voxelization is to define the bounding box of the region of interest. For most 3D shapes derived from point cloud areas, the most secure and controlled way to ensure complete coverage of this region is by using axis-aligned voxels. Otherwise, a larger number of voxels may be required to represent the same area. A reasonable voxel count is only guaranteed when the general orientation of the region's shape aligns with the voxel orientation. Once again, an increased number of voxels leads to longer computational times and a greater underestimation of the simulated illuminance values.

About the materials assigned to these voxels, two approaches were followed: the first based on the predominant object is included within each voxel, and the second on the point number inside each voxel. In the first approach, after constructing the voxel grid from the point cloud, each voxel was defined as branch/stem or leaf. This means that for the February representations, the voxels are classified as only branch/stem points, whereas for the June representations there are both of kinds of voxels.
On the other hand, regarding the second approach, followed by the creation of the voxel grid, the maximum number of points inside the voxel was defined. Then, for each voxel the following metric named transmittance index was computed:

$$\text{Transmittance Index} = \frac{\text{Max Point Number} - \text{Voxel Point Number}}{\text{Max Point Number}} \tag{3.1}$$

Then, four distinctive voxel grids out of the initial one were generated. More specifically, each voxel was integrated to one these four grids based on the transmittance index by taking the following cases:

1. If transmittance index ≥ 0.76, then this voxel belongs to the first voxel grid, meaning that the voxel is quite penetrable.
2. If $0.51 \leq$ transmittance index ≤ 0.75, then this voxel belongs to the second voxel grid.
3. If $0.26 \leq$ transmittance index ≤ 0.50, then this voxel belongs to the third voxel grid.
4. If transmittance index ≤ 0.25, then the voxel belongs to the fourth voxel grid. This indicates a high point density within the voxel, meaning that incident light from the simulation cannot easily penetrate it.

For the next two cases, the concept of the Density-based spatial clustering of applications with noise (DBSCAN) should be introduced. As discussed in the Nan's lecture notes [47], it is a density based clustering algorithm, which takes two parameters, a distance threshold, which defines the neighborhood of each point, and an integer specifying minimum number of points that needed to constitute a cluster. For each point in the dataset, it is checked whether at least this minimum number of points lies within the distance threshold. If so, the point and its neighbors are assigned to the same cluster. This point is defined as core point. Furthermore, points that do not individually meet the minimum requirement but are reachable from a core point through intermediate neighbors are also included in the cluster. Finally, points that are not reachable from any other point are classified as outliers (**Figure 3.2**).

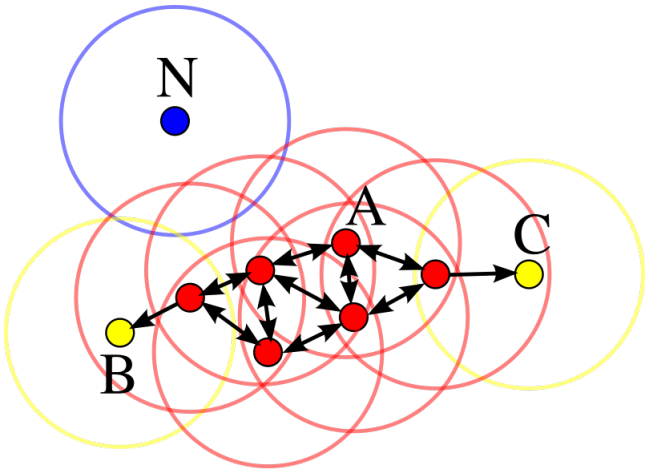


Figure 3.2.: Illustration of DBSCAN clustering. Core points (red) form a dense cluster, yellow points are not core point, but reachable from core points and point N (blue) is considered noise. Source: Chire [15].

Next, the process for the third case, the Convex Hull Case involves extracting the points representing the tree branches and constructing their convex hull. The first step was to select the points points with height lower than 1.5 m, followed by applying the DBSCAN algorithm to cluster them. After this step, the position of the trees is estimated automatically through the determination of the clusters. However, some clusters represent branches near the ground rather than tree roots (**Figures 3.3**).

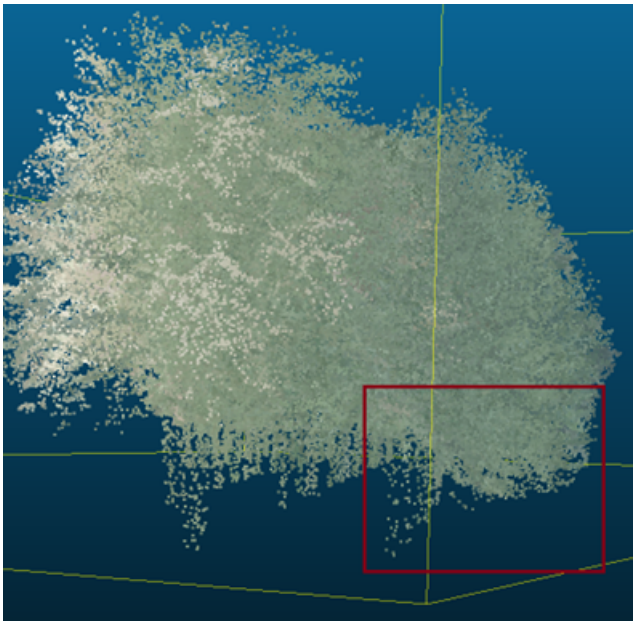


Figure 3.3.: The points within the red rectangle correspond to branches close to the ground, which are likely to form a cluster.

To address this, the "label" attribute was added to the low-points dataset and the data were segmented by unique labels. During execution, the clusters corresponding to non-tree elements were manually excluded. For each remaining segment (saved as an individual LAS file), an annulus was instantiated and translated to the coordinates $(\overline{x}, \overline{y}, z_{\text{min}})$. Subsequently, using the cleaned AHN 5 point cloud, every annulus was replicating along the Z-axis with a spacing of 1 m (empirically determined) until there was an annulus that contains points from the cleaned AHN 5 point cloud between its inner and outer ring. Only these new annuli which satisfied this condition were kept for the next step. (**Figure 3.4**).

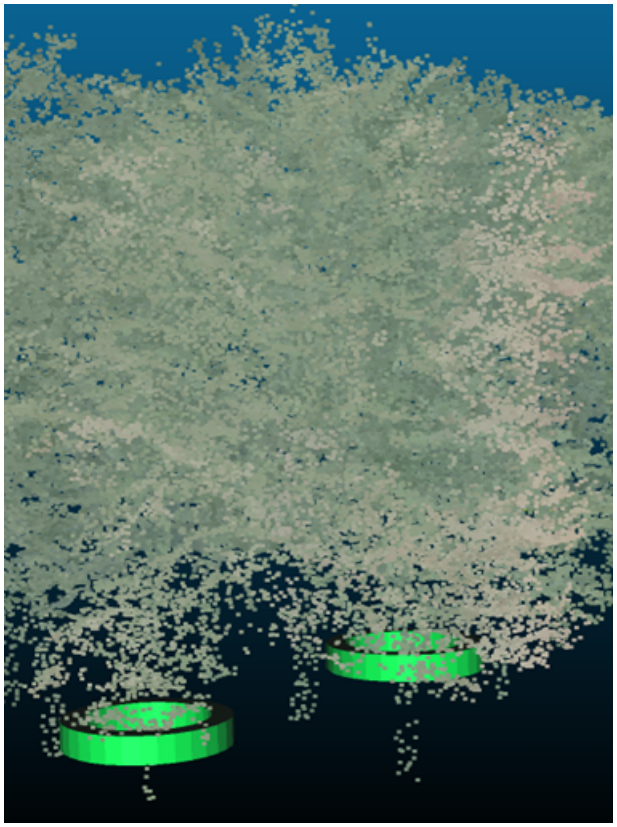


Figure 3.4.: Example of two trees with their annuli.

Then, from the cleaned AHN 5 point cloud, points with the following attributes related to the annuli remaining from the previous step were deleted: (**Figure 3.5**):

1. z coordinate lower than the lower height that is located inside the inner circle
2. x and y coordinates so that they are inside the inner circle.

Lastly, the remaining point cloud was used for the reconstruction of the convex hull.

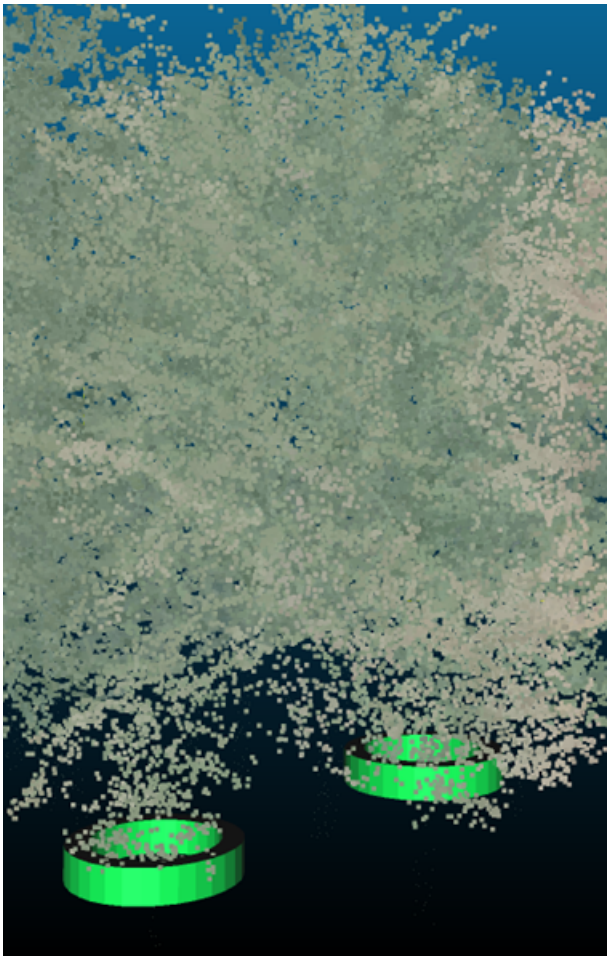


Figure 3.5.: Example of two trees after the deletion of their stem points.

The fourth case was the Alpha Shape Case, inspired by Zhu et al. (2008) [20]. The approach adopted in this case differs from the previous ones. More specifically, the cleaned AHN 5 point cloud was separated into individual trees, and the resulting LAS files were then used to generate the alpha shape from the points of each tree. To segment the clean AHN 5 point cloud, the DBSCAN algorithm was applied to the points with a height lower than 1.5 m like in the Convex Hull case. The lowest points of each cluster were then identified. Then, inspiration was drawn from Wang et al. (2021) [67], and therefore a 10-nearest neighbors network of the points was reconstructed. The choice of ten neighbors was based on experiments showing that this number avoids both disconnection and excessive complexity in the network. Based on the shortest distance calculations between the lowest points and all other points in the graph, points were classified into the tree to which they most likely belonged.

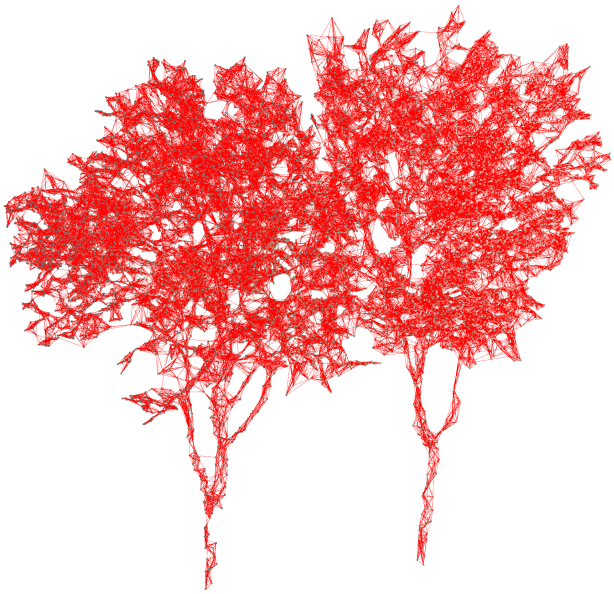
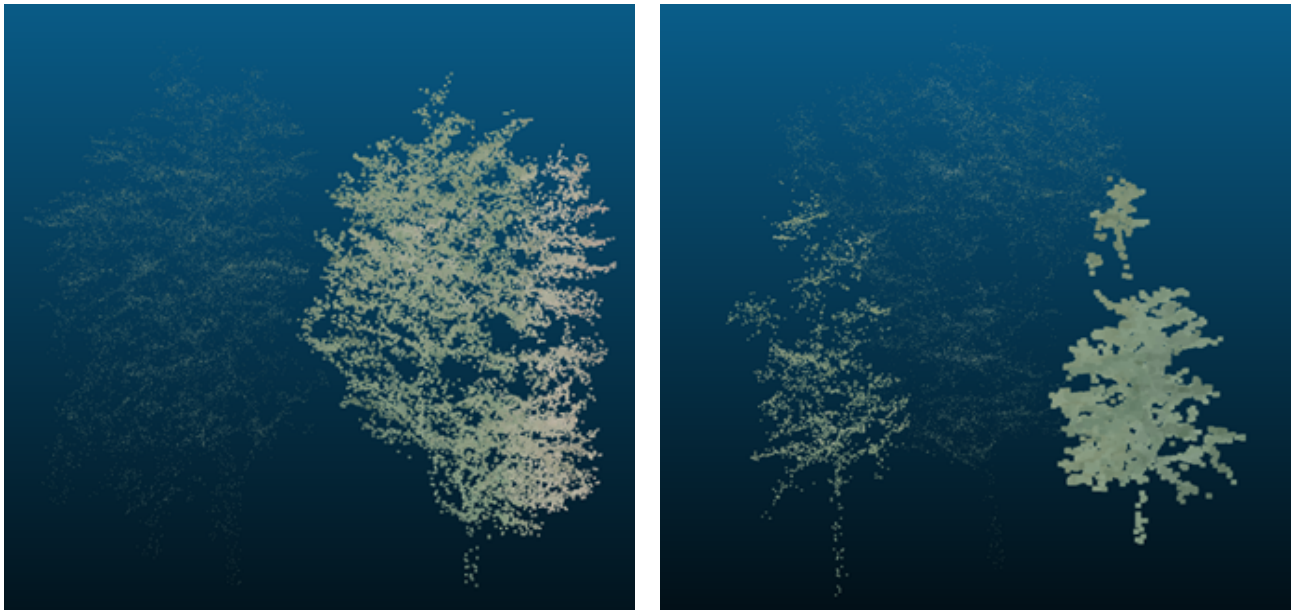


Figure 3.6.: Example of two trees interconnected by a 10-nearest neighbors network.

Although this algorithm significantly reduced processing time and achieved satisfactory results, its accuracy was insufficient, and the results therefore required manual refinement, since tree separation needed to be as precise as possible for this representation. Otherwise, the remaining errors would have caused problems in the implementation. **Figure** 3.7 shows both a successful **(a)** and a failed **(b)** tree separation. In general, severe misclassifications occurred when separating trees of different heights, as points from the far ends of long branches of a tall tree tended to merge with a smaller neighboring tree. This happened because, under the shortest-distance criterion, the lower points of the small tree were closer to those branches.



(a) Successful separation **(b)** Failed separation

Figure 3.7.: Examples of results from the tree separation implementation.

3.1.2. Daylight Simulation

The third step consisted of conducting the daylight simulation. As explained in Chapter 2, to run daylight simulation using the Daylight Availability workflow, location with solar data, materials of the scene and occupied areas should be defined. Considering the location of the study area, the most representative EnergyPlus Weather (EPW) file for the study area was defined among the embedded EPW files inside the Rhino software. To further adapt it to the local context, the Solar Radiation Data (SoDa) [55] platform was employed. This platform gives access to time series of GHI, DNI and DHI from CAMS solar dataset.
Then, after taking care for other parameters, that are related to the materials of the scene objects and surface on which the simulation is performed, the Daylight Availability workflow of Rhino software was used to produced hourly time series of daylight simulation throughout a year based on the new GHI, DNI and DHI values in the EPW file. As mentioned in Chapter 1, practically the simulations referred to only February and June, because the point cloud acquisition took place in February and in June 2023 the vegetation is quite dense. Therefore, after constructing the February representations using cleaned AHN 5 point cloud and the June representations with the same input point cloud data enriched by the additional leaf points, they were added at the scene with the georeferenced CCC Building model. Lastly, these combined scenes were used for the simulations. More details on how the tasks were done will be found in Chapter 4.

3.1.3. Simulation results evaluation

The fourth step was related to the evaluation of the results. For this task, it is assessed how accurately the simulation data fitted the reference, which is the actual illuminance sensor data. To implement this, after both February and June simulations, the corresponding simulated values were isolated and using the illuminance

records of February and June, the Root Mean Square Error (RMSE) and the Mean Bias Error (MBE) were calculated with the following mathematical structure:

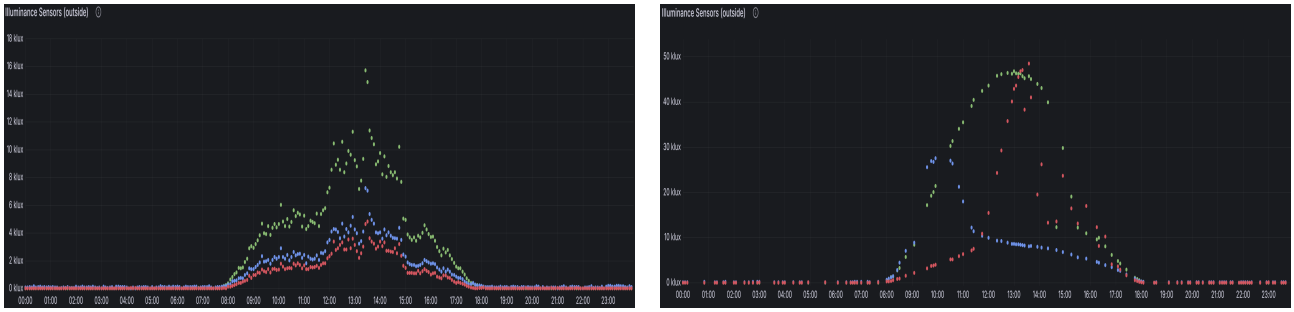
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.2}$$

$$\text{MBE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{3.3}$$

where:

- y_i = observed (measured) value,
- $\hat{y}_i$ = simulated (predicted) value,
- n = total number of observations.

These metrics set the basis of the simulation results assessment giving the initial evidence of which simulations performed better. The Voxel Grid Case, in which multiple configurations were evaluated by changing the voxel size and material, achieved good results in several instances for both months. To further assess the Voxel Grid Case and understand the factors contributing to these results, the nearest-voxel distance distribution was generated for each voxel size to examine how the spacing between voxels may influence the outcomes. Additionally, error distribution histograms for both months were constructed for each voxel size over the entire time series.
The next level of evaluation was to take five representative days of clear sky, cloudy and intermediate conditions, create three separate datasets and use them to make bar charts of hourly MBE values for each voxel size and material. For February, the hours which were taken into account were from 5:00 and 17:00 UTC (6:00 and 18:00 local time), since according to the *Sunrise and Sunset Times* website [62], sunrise and sunset in February occur approximately at 7:30 and 17:30 local time, respectively. Therefore, the selected hours for the diagrams range from 5:00 to 17:00 UTC (one hour before sunrise and after sunset) to ensure that all meaningful daylight hours are covered. For June, the same method was followed by including hours from 2:00 to 21:00 UTC in the bar graphs, as daylight lasts approximately from 5:30 to 22:10 local time.
For the selection of these days, the *Grafana* platform [59] was used, on which data of the sensors from the CCC building are published in order to select the days of the three categories based on the global horizontal illuminance graph. This sensor is facing the zenith and, therefore, it can be understood when and how long there was cloud coverage. At **Figures 3.8** to **3.9**, time series graphs of the records of global horizontal, east and west sensors in green, blue and red respectively, for a clear sky and cloudy day in February and June 2023. On cloudy days, the illuminance values are lower and, at the same time, there are intense fluctuations and disruptions for global horizontal, in case the cloud coverage is not consistent, while the other graphs are more uniform and tend not to be distinguishable from each other. In contrast, under clear-sky conditions, the global horizontal graphs show no disruptions and tend to follow a normal distribution pattern, while the other two diagrams also remain continuous and appear as mirror curves due to their different orientations. Interestingly, during clear sky days, particularly in June, it is evident that the CCC building is shaded by the surrounding trees, as the illuminance values recorded by the sensors are lower even in the absence of clouds.



(a) Cloudy day in February 2023 (21/2/2023) **(b)** Clear sky day in February 2023 (14/2/2023)

Figure 3.8.: Global horizontal (green dots), east (blue dots) and west (red dots) illuminance sensor graphs (values in klux) for a cloudy day and a clear sky day in February 2023 from Grafana platform.

(a) Cloudy day in June 2023 (22/6/2023) **(b)** Clear sky day in June 2023 (14/6/2023)

Figure 3.9.: Global horizontal (green dots), east (blue dots) and west (red dots) illuminance sensor graphs (values in klux) for a cloudy day and a clear sky day in June 2023 from Grafana platform.

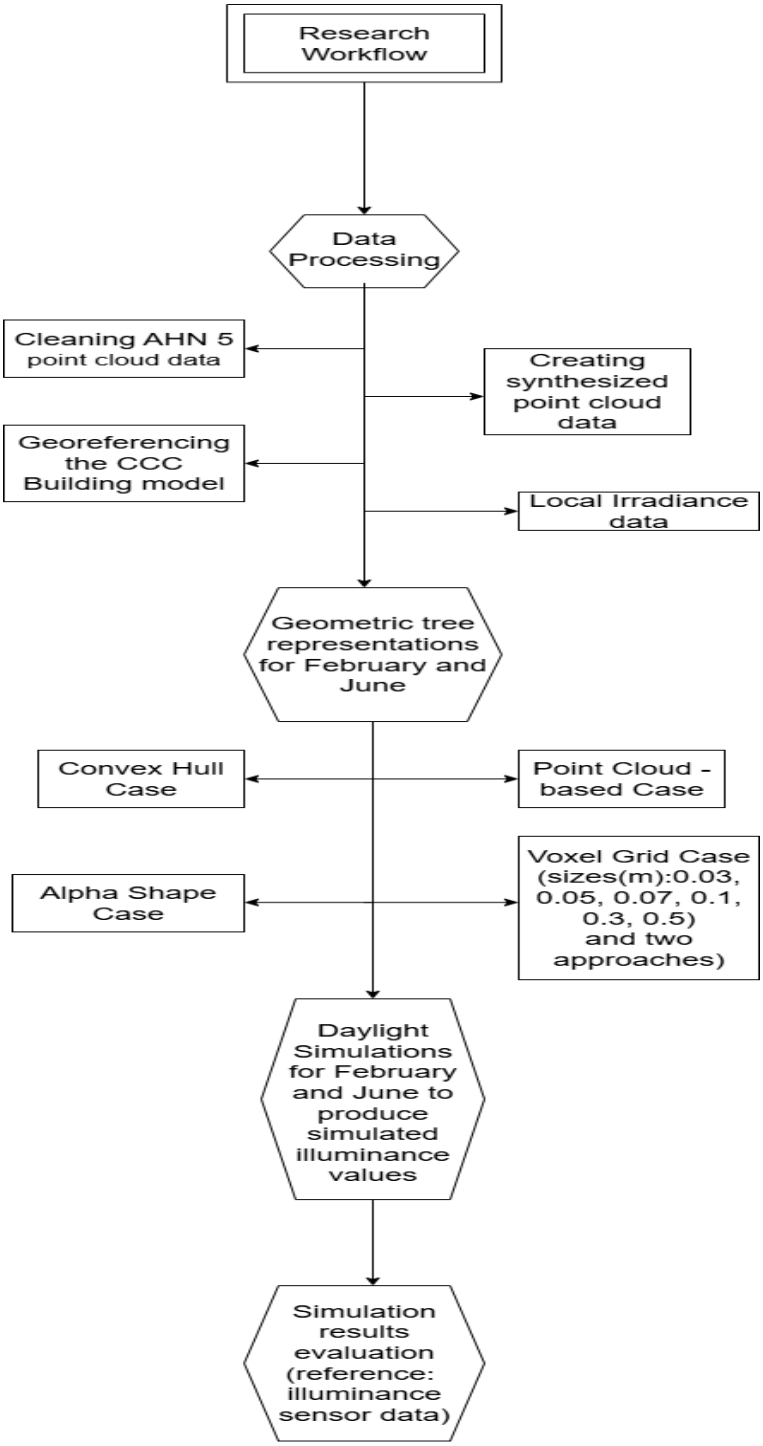


Figure 3.10.: Research workflow including its four steps: Data Processing, Geometric tree 3D representations, Daylight Simulation, Simulation results evaluation

4. Technical Implementation

In this section technical details regarding the tools and data used for the generation of the geometric representations,the georeferencing of the CCC Building 3D model, the procedures for generating the geometric representations of the trees and the daylight simulation in this research.

4.1. Tools

For each of the geometric representations analyzed in the previous chapter, an executable in Python 3.11 was constructed. The included libraries and their purposes are outlined at **Table 4.1**.

Table 4.1.: Python libraries and their purposes in this work

Library	Purpose
laspy	Reading and writing LAS/LAZ point cloud files
numpy	Numerical operations and array handling
open3d	Voxel grid construction and point cloud processing
scikit-learn	Density-based clustering (DBSCAN)
trimesh	Geometric transformations and convex hull construction
alphashape	Alpha shape generation
shapely	Geometric operations and topology checks

Apart from Rhino, CloudCompare and MeshLab are used for point cloud processing (e.g. sampling leaf points, extracting point cloud density) and visualization. Also, QGIS was deployed for visualization of 2D geospatial data and AutoCAD was used to create the scenes for the daylight simulations.

4.2. Data

The spatial data already obtained the AHN 5 point cloud LAZ file, `37EN2_16.LAZ`, which includes the study area and was downloaded from Geotiles.nl [31]. This platform was chosen because it provides an additional tiling scheme, subdividing each original AHN tile (5 × 6.25 km) into 25 smaller tiles of 1 × 1.25 km. Since the study area covers only 3,900 m², a single subtile from Geotiles is sufficient and simplifies the processing. Based on the datasets of flight lines and their corresponding scanned areas provided by Dataroom [3] these point cloud data were acquired in February 2023.
Moreover, a DWG file, that contains a highly detailed representation of the CCC Building, a simplified and abstract representation of the trees from the nearby tree area and the surrounding area was supplied by the Green Village (**Figure 4.1**) for the processing steps.

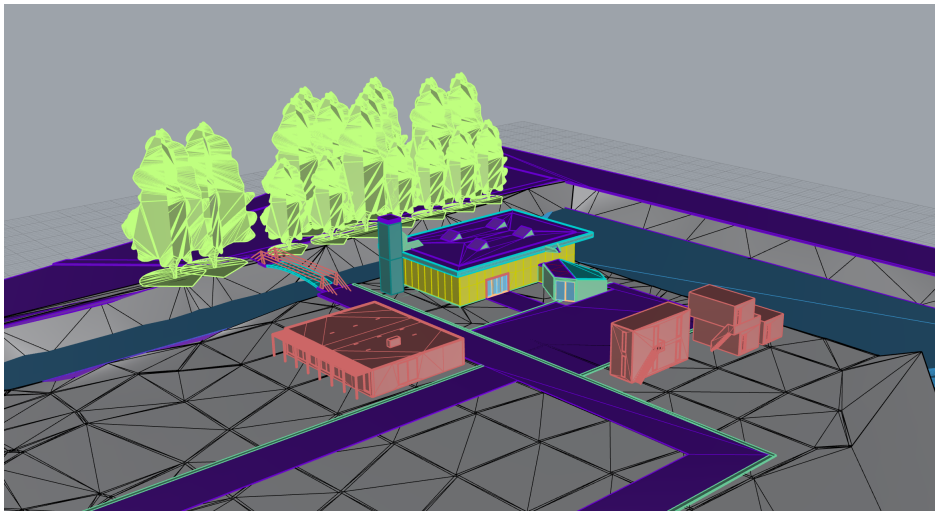


Figure 4.1.: Scene of study area in Rhino

For the step of georeferencing the initial 3D representation of the CCC Building, the 3D BAG LOD 2.2 model of CCC Building derived from 9-288-552.gpkg [40] were utilized. It is uploaded as OBJ and geopackage file, the first of which then converted to DWG file through Rhino software, in order to serve the horizontal georeferencing through AutoCAD.
Furthermore, the illuminance records from west sensor of the CCC Building for February and June 2023 were obtained from the university staff responsible for managing this data. In **Figure 4.2** and **4.3**, the time series of the corresponding hourly records in UTC time can be seen.

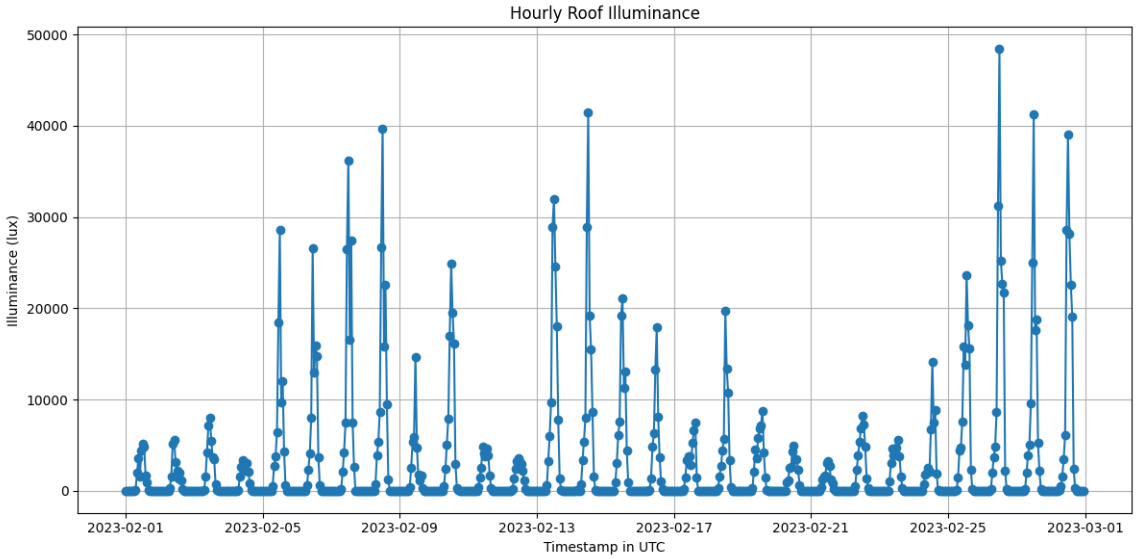


Figure 4.2.: Time series hourly illuminance values of February from the west sensor of the CCC Building

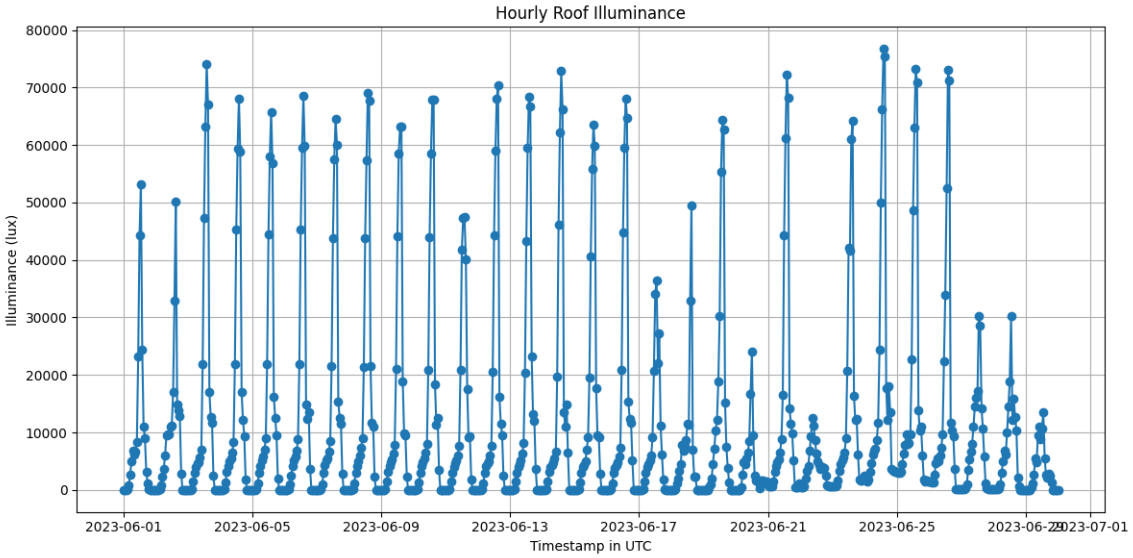


Figure 4.3.: Time series hourly illuminance values of June from the west sensor of the CCC Building

Finally, complementing what was described in Chapter 3, to adjust the EPW file in order to reflect the study area practically this EPW file was modified by replacing the default values of its fields of the GHI, DNI, and DHI with the corresponding values of the from the weather file of year 2023 from the SoDa platform.

4.3. CCC Building georeference

Based on the information from 3D BAG webpage ([2] and [1]) at **Figure 4.4**, the Root Mean Square Error (RMSE) of the 3D distances between the AHN 5 point cloud and the LOD 2.2 model of CCC Building is **0.141** m.

Figure 4.4.: LOD 2.2 model of the CCC Building from the 3D BAG website and its attributes, including b3_RMSE_lod22

Due to the inconsistency between the detailed 3D model of the CCC Building in the initial scene and the corresponding LOD 2.2 model, the horizontal georeferencing could only be based on the edges of its small part, since the roof of this part does not have eave (**Figures** from **4.5** to **4.6**). The initial coordinates of the six vertices that define the base of the small part of the CCC building are presented at **Table** 4.2.

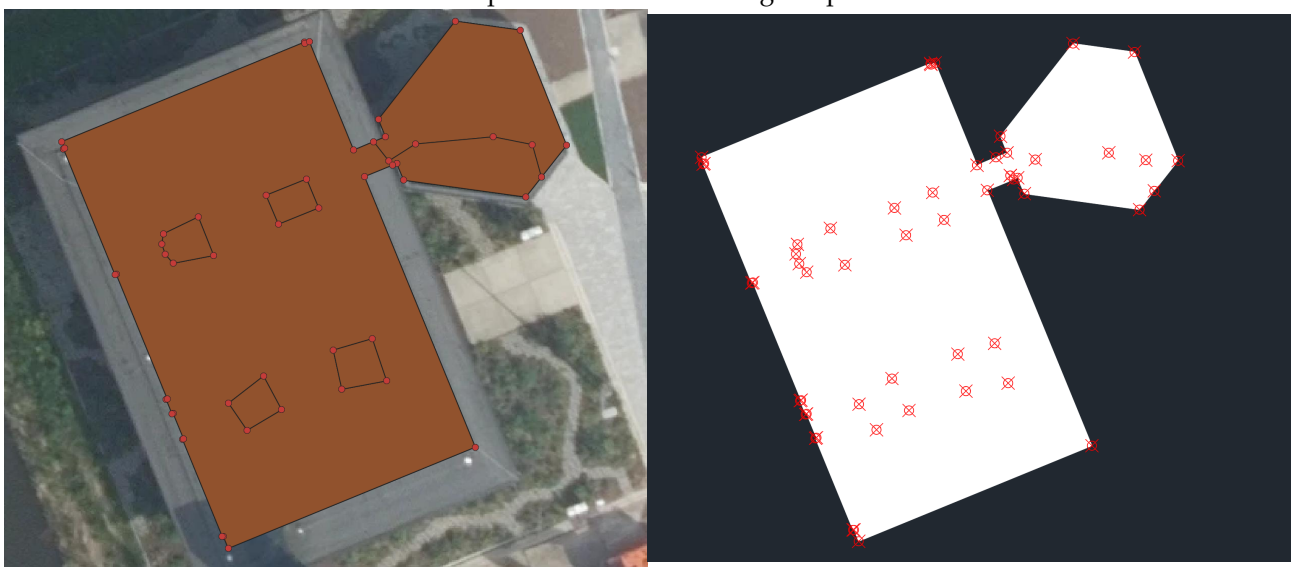


Figure 4.5.: LOD 2.2 of CCC Building with its vertices of the outlines in all levels in QGIS and AutoCAD.

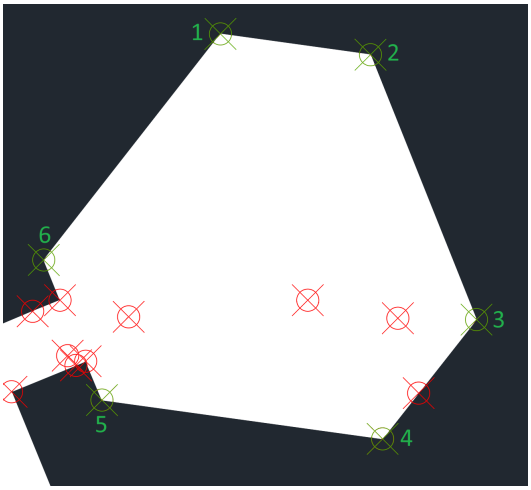


Figure 4.6.: Six vertices used for the horizontal georeferencing

Table 4.2.: 2D coordinates of the six vertices

	X	Y
1	85646.398	445865.875
2	85649.781	445865.406
3	85652.172	445859.438
4	85650.055	445856.750
5	85643.727	445857.625
6	85642.414	445860.781

After translating and rotating the CCC building model, the final positions of the six corresponding vertices are demonstrated in **Table 4.4**. The final alignment of the selected portion of the CCC building, along with the vertices from the 3D BAG model used for horizontal georeferencing, is illustrated in **Figure 4.7**.

Table 4.3.: Final 2D coordinates of the six corresponding vertices in the scene

	X	Y
1	85646.417	445865.748
2	85649.675	445865.296
3	85652.042	445859.47
4	85650.022	445856.875
5	85643.793	445857.738
6	85642.555	445860.784

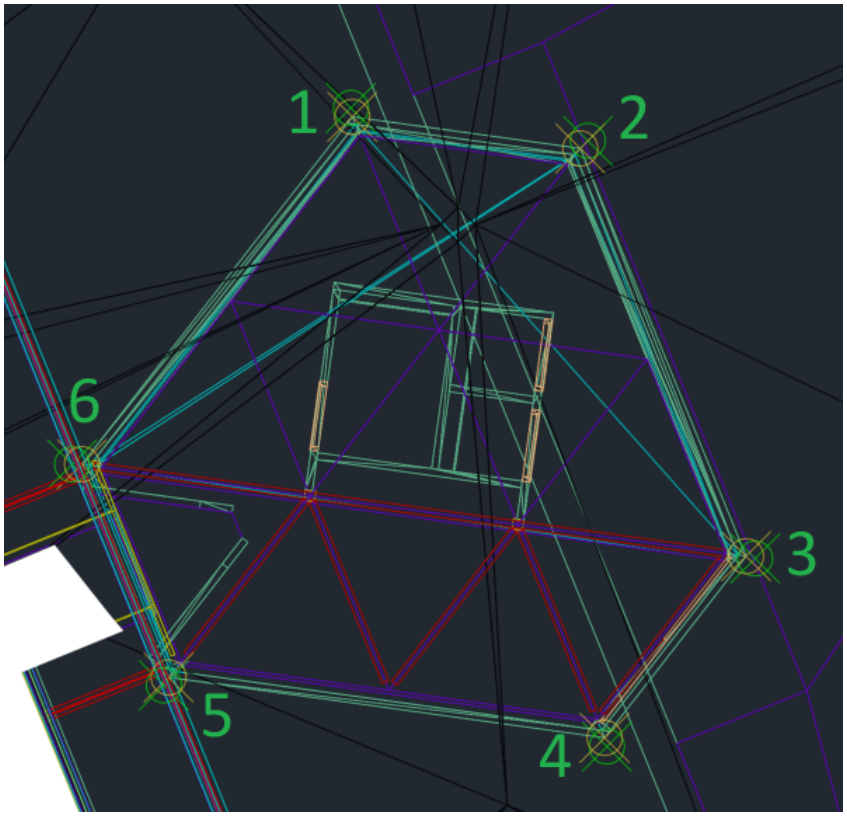


Figure 4.7.: Final position of the small part of the CCC building (3D BAG coordinates in green and final positions of the corresponding vertices in yellow)

This georeferencing is acceptable, as the differences between the coordinates from the 3D BAG model and those of the corresponding vertices are minimal. Moreover, the (RMSEx, (RMSEy and combined RMSE values presented at **Table** 4.4) do not exceed the RMSE of the 3D distances between the point cloud and the LOD 2.2 model of the CCC Building.

Table 4.4.: RMSE_x, RMSE_y **and** RMSE **in meters**

$\text{RMSE}_x(m)$	$\text{RMSE}_y(m)$	$\text{RMSE}(m)$
0.095	0.098	0.136

The vertical georeferencing was facilitated by the main part of the CCC Building, because the roof of the previous part of the building is modelled in a more complicated way and, therefore, it cannot be related to the much more simplified roof of this part of the 3D BAG building. On the other hand, the height of the CCC Building can be adjusted using the flat part of the roof of the 3D BAG model, that has height of 5.226 m, excluding its four windows.

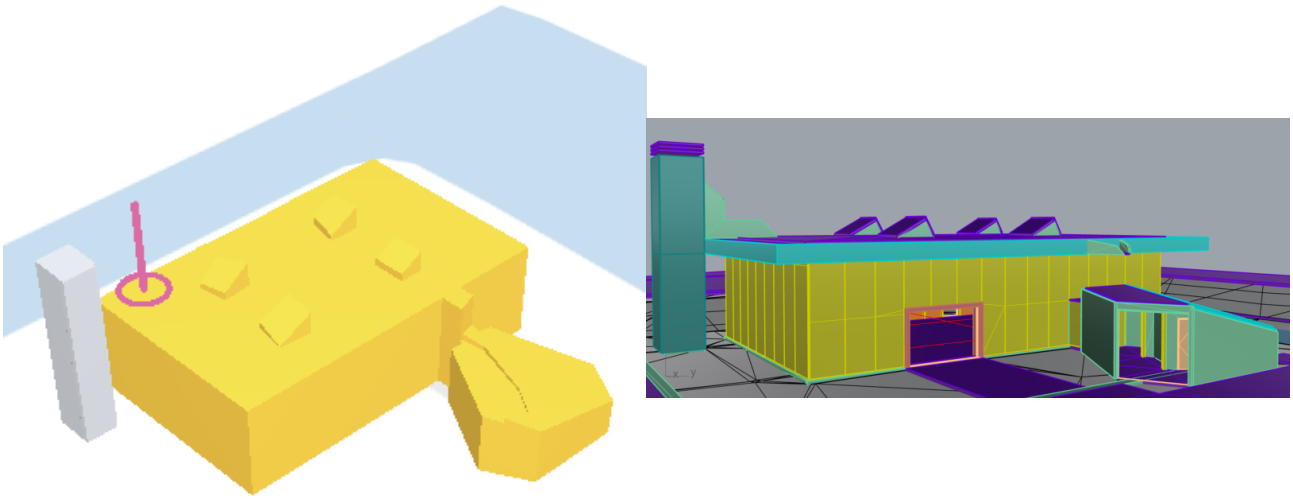


Figure 4.8.: Screenshots of CCC Building from 3D BAG and Rhino.

Finally, the entire CCC building model was vertically shifted inside Rhino software so that its flat largest roof surface positioned at 5.226 m (**Figure 4.9**).

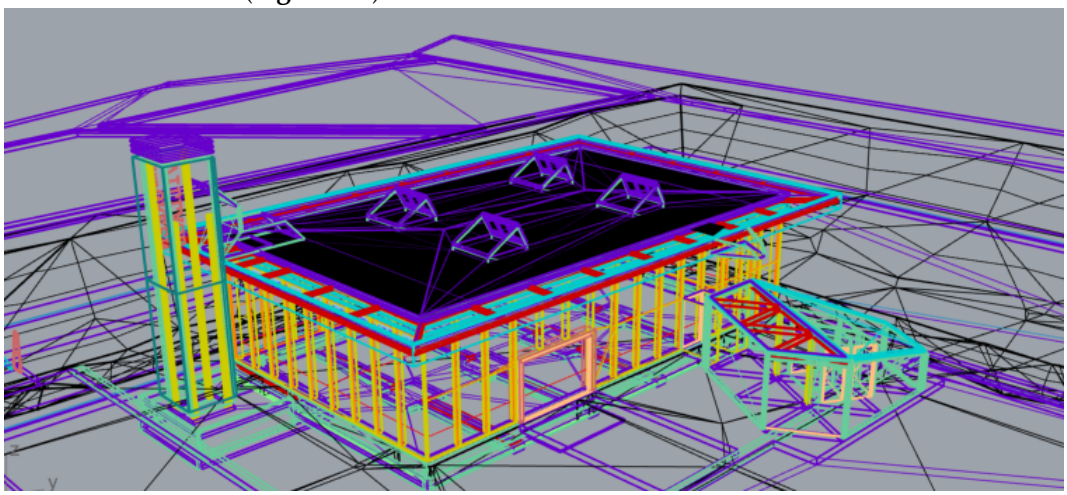


Figure 4.9.: CCC Building and the upper surface of the 3D BAG model

The ability to assess the position in the z-direction was limited. However, the height differences between the two points shown in **Figure 4.10** and **Figure 4.11** were determined to be 0.116 m and 0.255 m, respectively.

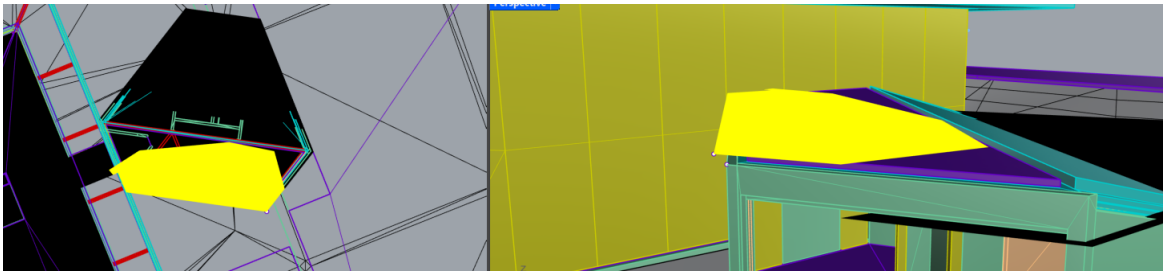


Figure 4.10.: Points to evaluate the final height of small part of CCC Building

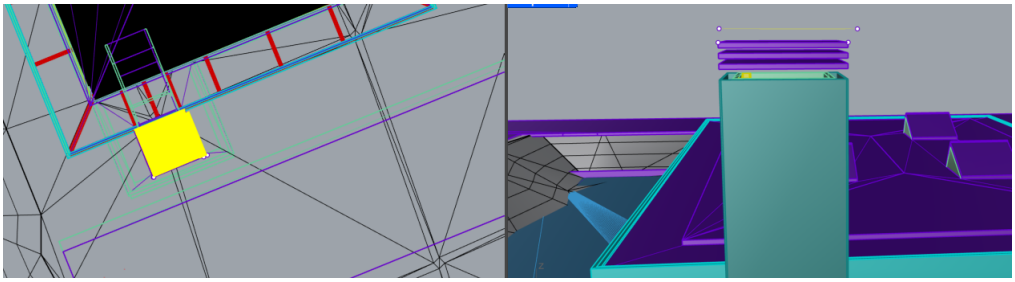


Figure 4.11.: Points to evaluate the final height of chimney

Despite the large discrepancy at the points shown in **Figure 4.11**, the final position may still be acceptable. This is because the 3D BAG model representing the chimney is separated from the CCC Building, and precise alignment between the top of the Rhino chimney model and the corresponding 3D BAG geometry is less critical than it is for the roof of the smaller adjoining structure.

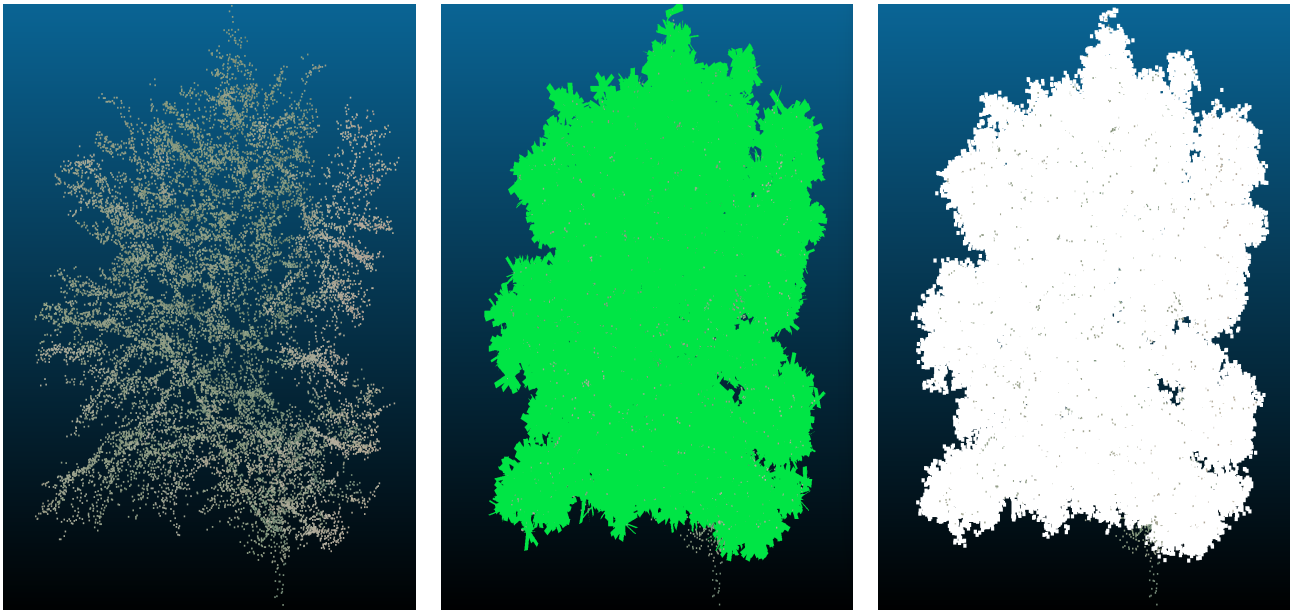
4.3.1. Technical details of geometric tree representations

As previously mentioned, the geometric tree representations considered in this study include the point cloud-based, voxel grid, alpha shape, and convex hull representations. For geometries referring to February, the input was the AHN 5 point cloud, cleaned by removing points not belonging to the trees. In contrast, for the June geometric representation the AdTree algorithm [22] was employed. Specifically, after segmenting the tree point cloud into individual trees, synthesized leaves were generated for each tree using the algorithm and were saved in OBJ format. Subsequently, point sampling was applied to each OBJ file of the synthesized leaves in CloudCompare using the Edit > Mesh > Sample points function, with the configuration set to 30 points per square unit (**Figure 4.12**). Statistics of each point cloud are presented at **Table 4.5**. It should be noted that the inclusion of more points was not feasible due to the computational limitations of the available system.

Table 4.5.: Statistics of the cleaned AHN 5 and synthesized point cloud

	Cleaned AHN 5 Point Cloud	Synthesized Point Cloud
Point Number	562,677	1,248,675
Mean Point Number per square meter	14.7 $\pm$ 5.9	35.1 $\pm$ 16.3

Generally, all the tree 3D representations were saved in OBJ format after completion of the process of each executable. Then, via Rhino software each of them was converted into DWG file, transferred into AutoCAD



(a) Individual tree point cloud. **(b)** Synthesized leaves from AdTree. **(c)** Sample leaf points.

Figure 4.12.: Example of a tree point cloud from dataset and sample leaf points derived from the synthesized leaves.

together with georeferred CCC Building model creating a combined scene in DWG format. Finally, these combined scenes from all the cases were utilized for the simulations.

4.3.2. Point Cloud - based Case

As mentioned in Chapter 3, to make points detectable in Rhino software, when using Climate Studio Plugin, each point was turn into a cube of edge length of 0.02 m. For the February geometric representation, since at that time period trees are bare, only the cleaned AHN 5 point cloud was used, while for the June representation both the cleaned AHN 5 point cloud data and the point cloud with only the leaf sample points were the inputs. All these cubes are saved at an OBJ file for February and at two OBJ files for June. At **Figure 4.13** the February geometrical representation of this case can be seen from two perspectives.
With respect to the materials assigned to the cubes for the daylight simulations in Rhino, two options were considered. For the February geometric representation, a single opaque material from the Rhino database was used, with a reflectance of 35.9 %, representing wood derived from a maple tree. For the June geometric representation, in addition to this wood material, another opaque material with reflectance of 13.7 %, was applied to represent the leaf points of the synthesized point cloud. The flow chart of the tasks for the processing in this case is demonstrated at **Figure 4.14**.

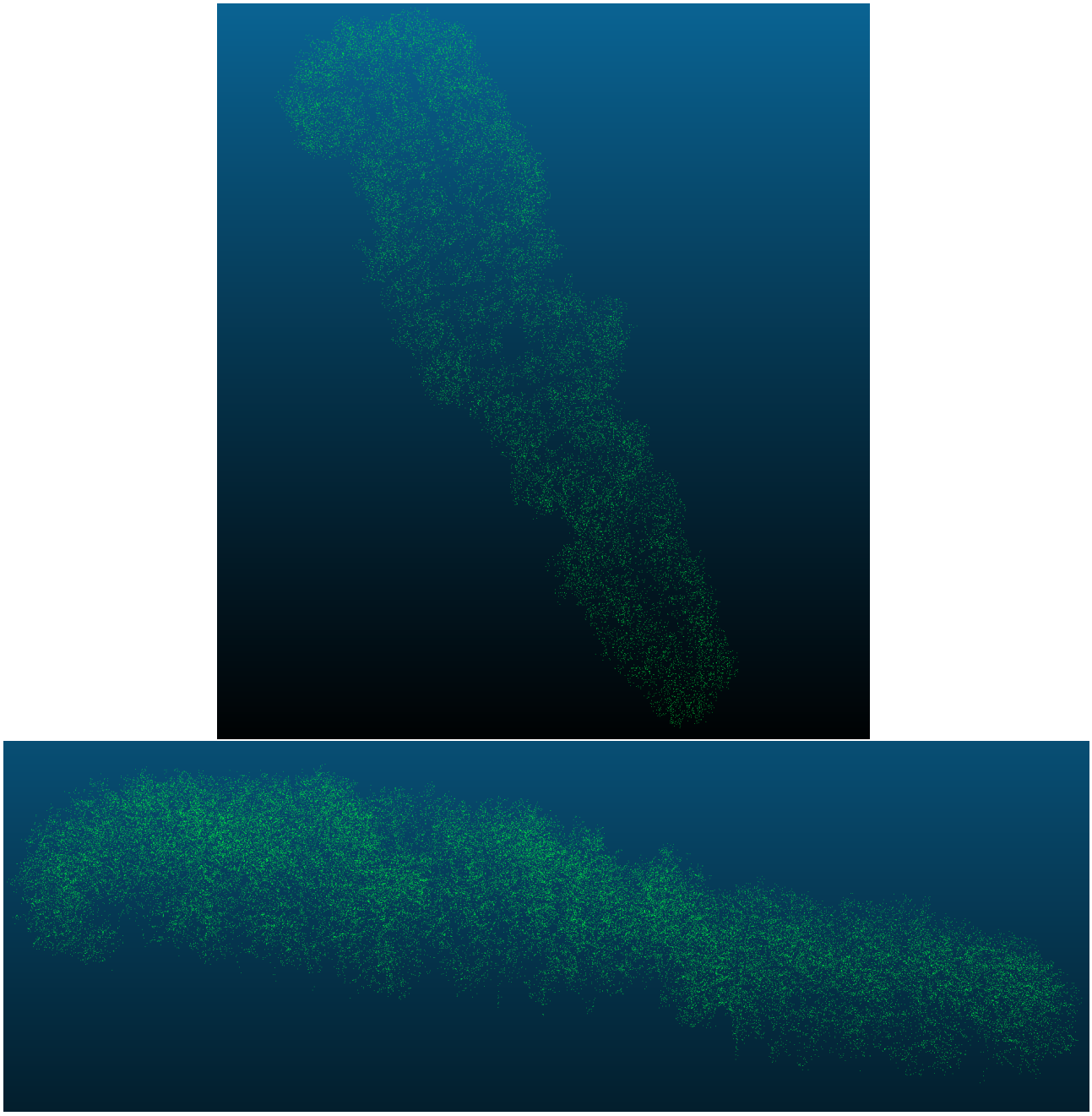


Figure 4.13.: Geometry resulting from the Point Cloud Case processing in February.

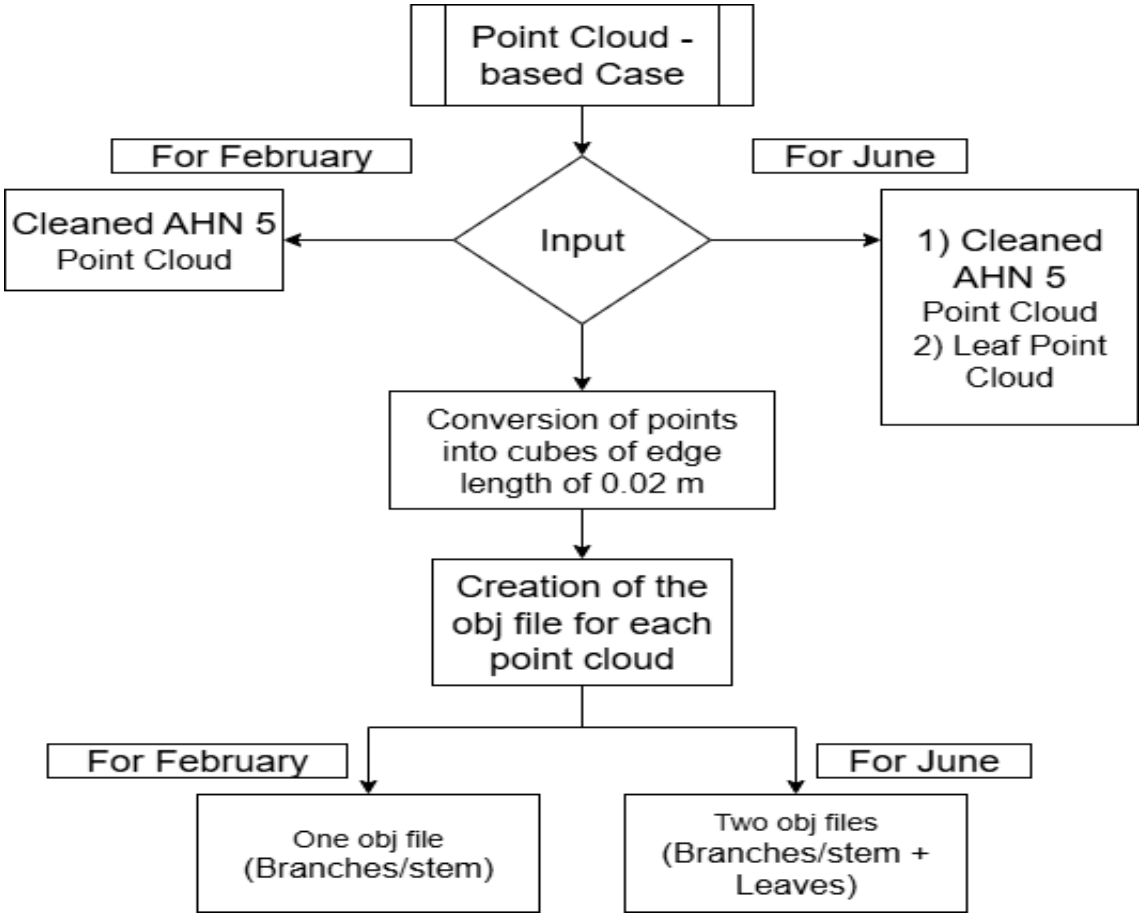
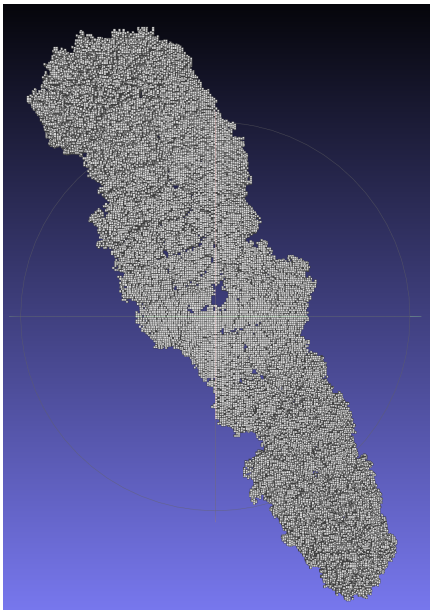


Figure 4.14.: Flow chart of the Point Cloud Case approach.

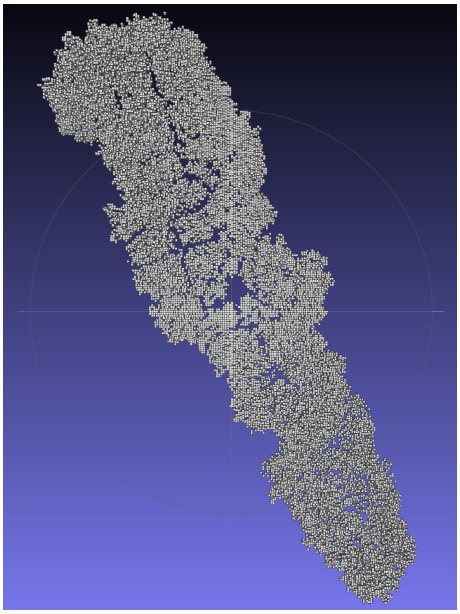
4.3.3. Voxel Grid Case

Regarding the Voxel Grid Case, after constructing the voxel grid from the point cloud using the `open3d` library, the voxel grid was separated depending on the approach which was followed. For the first approach, for the June representation, the algorithm received both the cleaned AHN 5 point cloud data and the point cloud with only the leaf sample points, created a unified voxel grid from these two point clouds and then each voxel was examined to determine whether it contained more branch/stem points (from the cleaned AHN 5 point cloud) or leaf points (from the leaf point cloud), and was assigned either the aforementioned wood or leaf material accordingly. For the February representation, since only the cleaned AHN 5 point cloud was used as input, all voxels were assigned the wood material. For February, there was one OBJ file, while for June two OBJ files, for branches/stem points and leaf points.
On the other hand, for the second approach, the input point cloud for February voxel grid was the same as the first approach, while for June voxel grid the whole synthesized point cloud was used. Then, the separation of the initial constructed voxel grid into four grids was implemented according to its transmittance index, as explained in Chapter 3. **Figure 4.15** presents the four voxel grids corresponding to the different transmittance categories, each with a voxel size of 0.50 m.
To ensure consistency with the Rhino software settings, four text files were created to define the glass material

with customized transmittance. The transmittance values in these files were set to the mean values of the intervals 0–0.25, 0.25–0.50, 0.50–0.75, and 0.75–1.00 (**Figure 4.16**). For both February and June, four OBJ files were saved corresponding to each transmittance value. The process according to each approach is summarized at **Figures 4.17** and **4.18**.

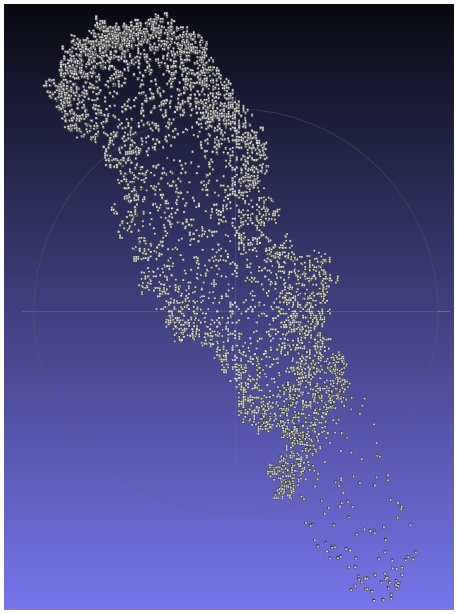


(a) Voxels from the grid with a size of 0.50 m and a transmittance index of ≥ 0.76.

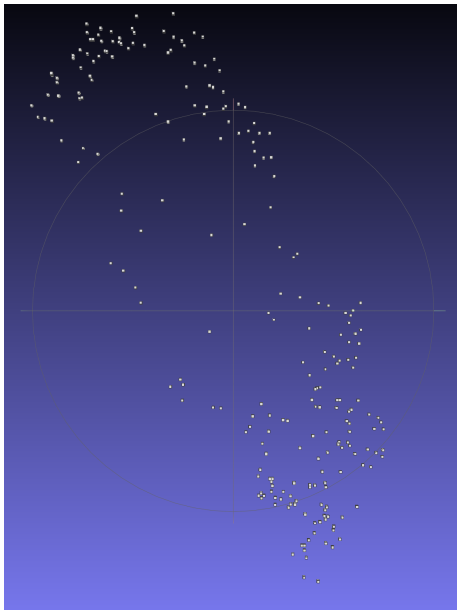


(b) Voxels from the second grid with a size of 0.50 m, where $0.51 \leq$ transmittance index ≤ 0.75.

Figure 4.15.: Example of voxel grid implementation with a voxel size of 0.50 m and two different transmittance index ranges derived from the February tree point cloud. (continued on next page)



(c) Voxels from the grid with a size of 0.50 m, where 0.26 $\leq$ transmittance index $\leq$ 0.50.



(d) Voxels from the grid with a size of 0.50 m and a transmittance index of $\leq$ 0.25.

Figure 4.15.: Example of voxel grid implementation with a voxel size of 0.50 m, illustrating the rest of transmittance index ranges derived from the February tree point cloud.

```
void glass First
0
0
3 0.88 0.88 0.88
```

(a) Voxel material of the first grid.

```
void glass Second
0
0
3 0.63 0.63 0.63
```

(b) Voxel material of the second grid.

```
void glass Third
0
0
3 0.38 0.38 0.38
```

(c) Voxel material of the third grid.

```
void glass Forth
0
0
3 0.125 0.125 0.125
```

(d) Voxel material of the fourth grid

Figure 4.16.: Voxel glass materials with customized transmittance values.

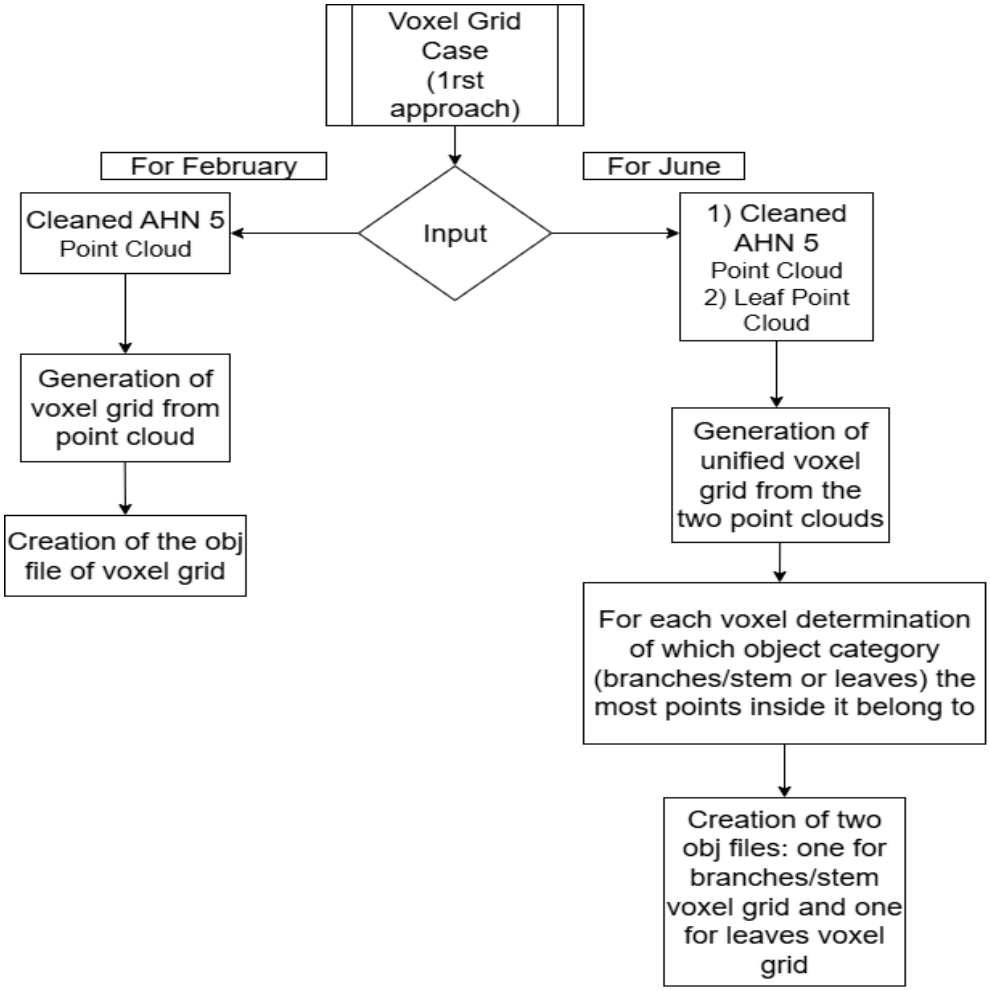


Figure 4.17.: Flow chart of the first Voxel Grid Case approach.

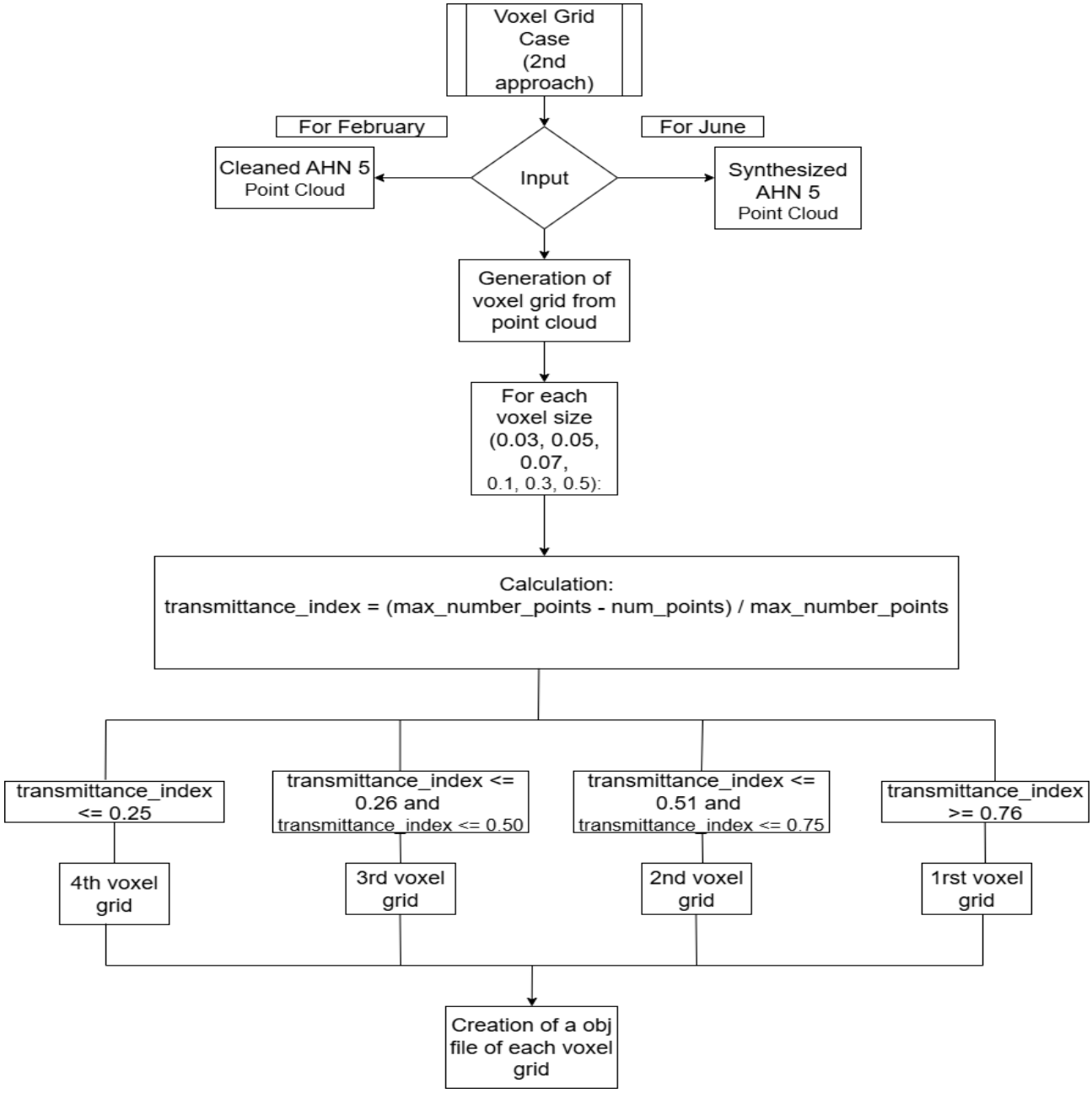


Figure 4.18.: Flow chart of the second Voxel Grid Case approach.

Regarding the processing time, the Voxel Grid Case was the most time-consuming. To accelerate the execution of the program, the built-in multiprocessing module was used, which enables multiple processes to run in parallel on separate Central Processing Unit (CPU) cores. **Figures 4.19** to **4.21** present the processing time in seconds per voxel size for the first approach in June, as well as for the second approach in both February and June. These processing times may vary when executed on a different computer or under different system conditions. No plot is provided for the first approach in February, as no complex processing was performed

in this case.
In overall, it is apparent that the processing time values of February are much lower than in the both approaches of June, due to the lower number of points of the cleaned AHN 5 point cloud compared to the synthesized point cloud. Also, the common elements of all the diagrams from 0.10 m onward the processing time drops dramatically. A similar trend would normally be expected for sizes below 0.10 m; however, due to the higher processing complexity associated with the larger number of voxels, combined with potential fluctuations in system conditions, the results in this range are less predictable.

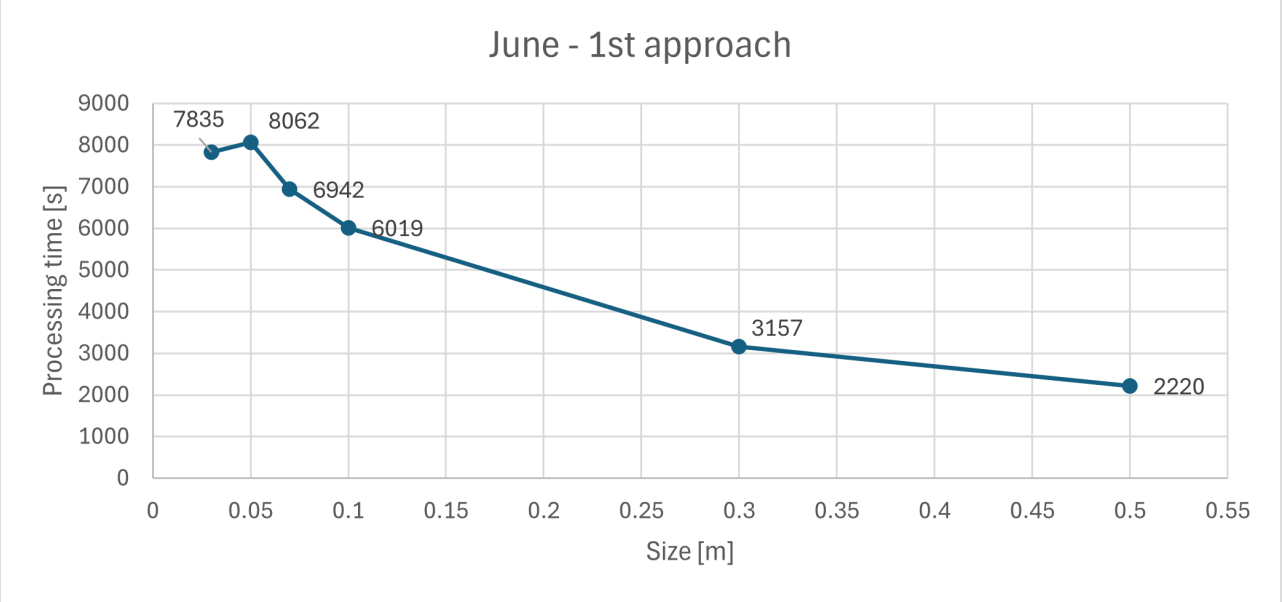


Figure 4.19.: Processing time in seconds per voxel size for the first approach in June.

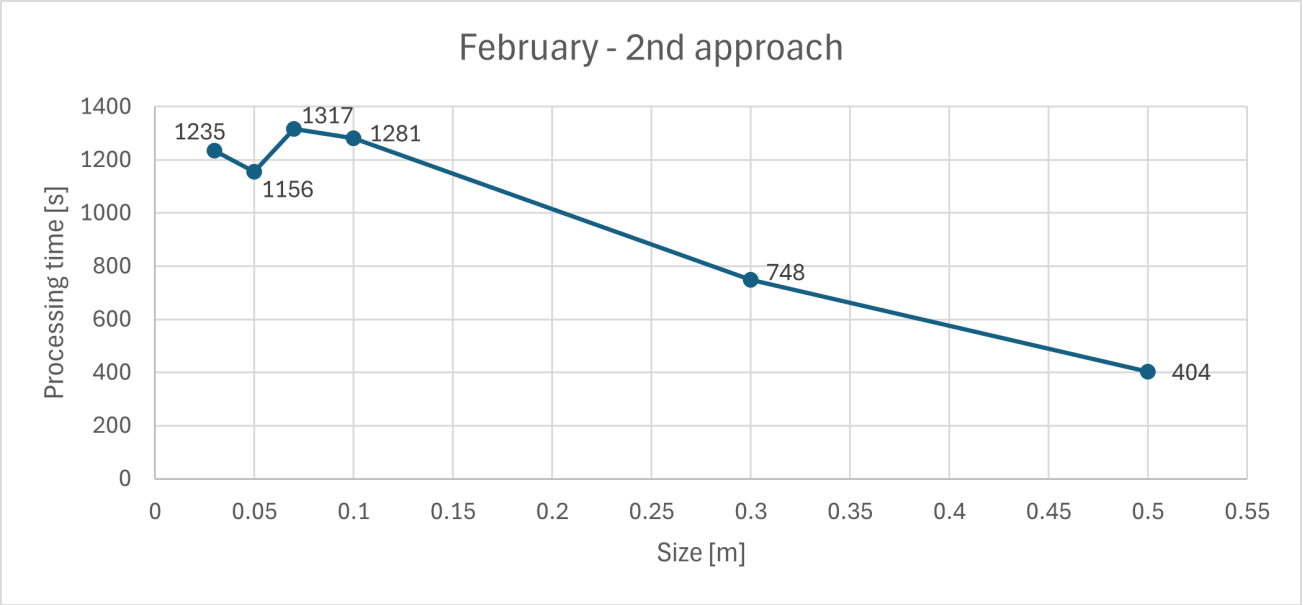


Figure 4.20.: Processing time in seconds per voxel size for the second approach in February.

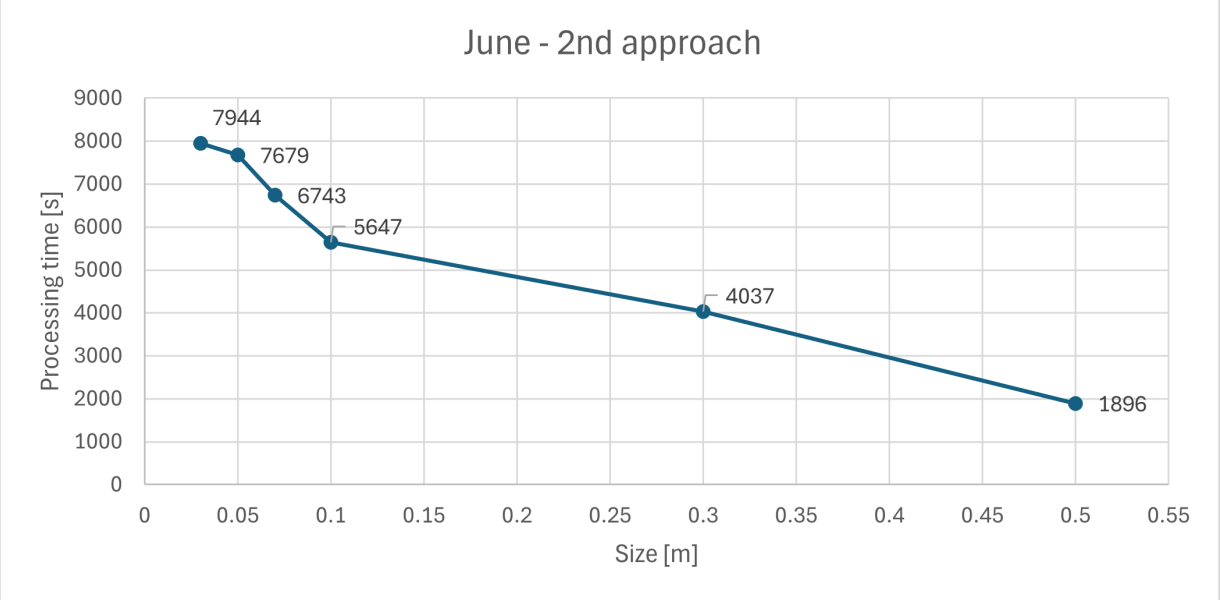


Figure 4.21.: Processing time in seconds per voxel size for the second approach in June.

4.3.4. Convex Hull Case

For the convex hull construction of the branches several parameters were determined empirically through experimentation. At first, to apply the DBSCAN algorithm in order to cluster the points with height lower than 1.5 m using the parameters: **radius = 0.8** and **minimum points = 3** With this step, the position of the trees was determined. However, there are some clusters in the northeast area that do not represent roots of trees, because there the trees are inclined and as a result some of their branches are close to or touch the ground (**Figure 4.22**).

After removing the outlier cluster, as explained in Chapter 3 parameters of the annulus created for all the remaining segments and used to remove the stem points were the following: **inner radius = 1.5, outer radius = 2.0, and height = 0.8** (empirically determined). Lastly, the convex hull generated from the remaining points that represent the branches was saved in OBJ format for both February and June. In terms of material assignment, the same wood material used in the previous cases was applied exclusively in the February representation, while in the June representation the convex hull object was assigned the same leaf material. At **Figure 4.24** the whole process for this case is outlined.

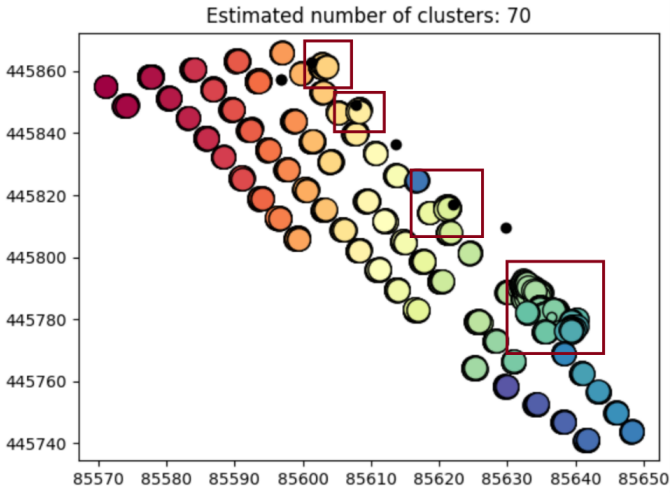


Figure 4.22.: The clusters highlighted within the red rectangles do not represent individual trees but rather branches extending close to the ground. The black points indicate outliers.

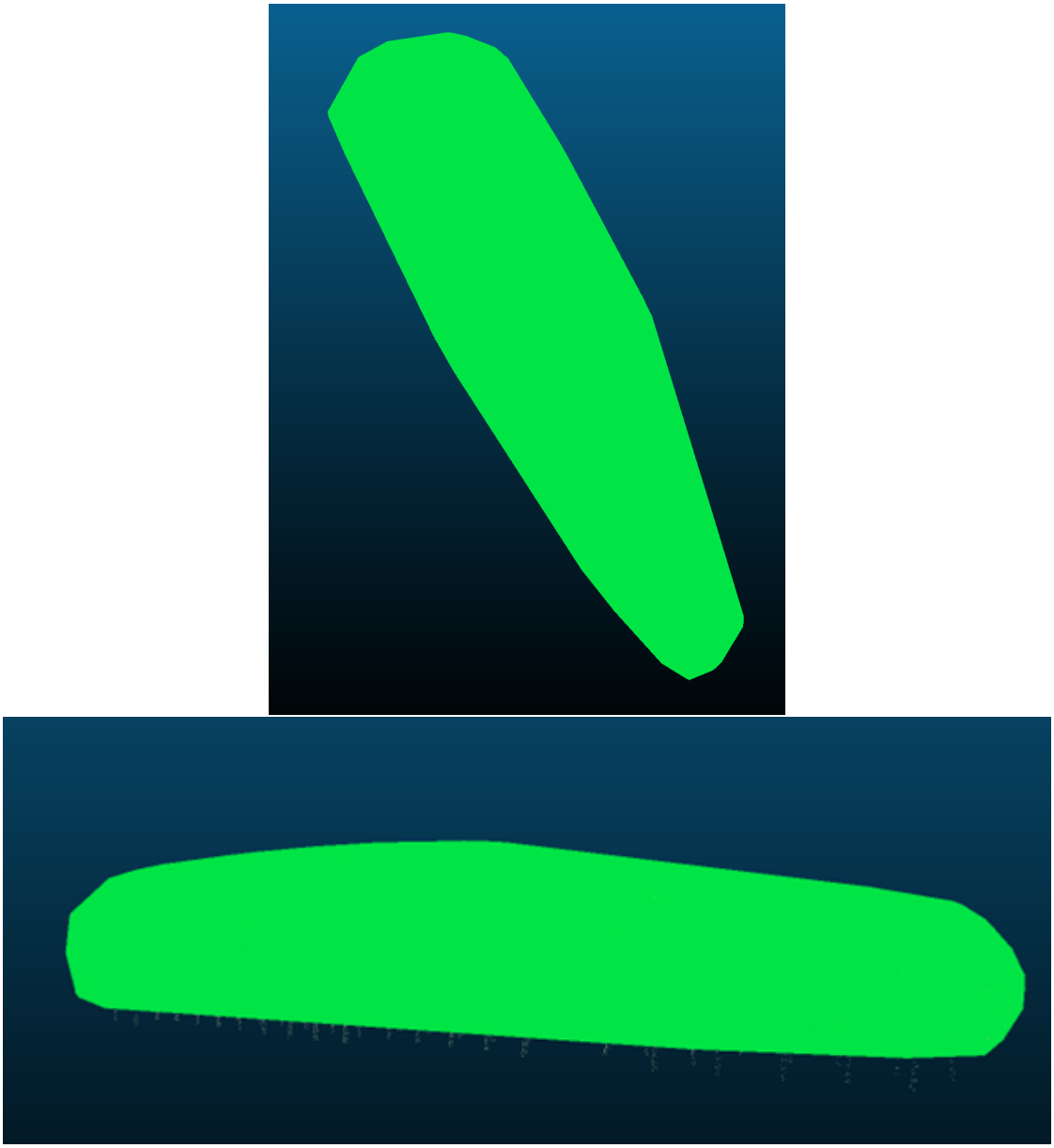


Figure 4.23.: Convex hull of the branch points of the February representation from two perspectives.

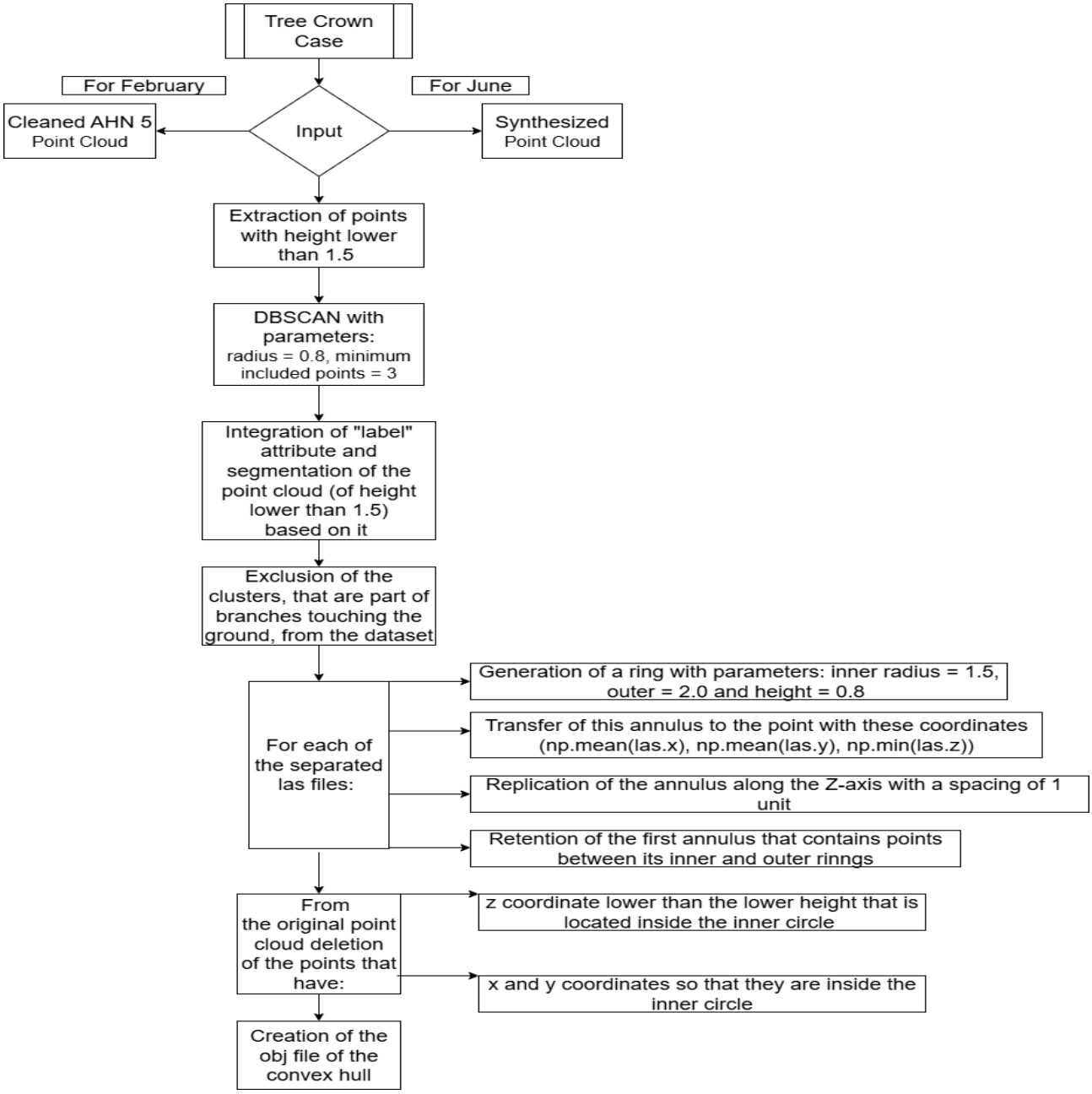


Figure 4.24.: Flow chart of the Tree Crown Case.

4.3.5. Alpha Shape Case

In continuation with what was mentioned in Chapter 3, after obtaining the individual tree LAS files, the alpha shape algorithm was applied to each of them and the final output was saved in OBJ format. It is important to note that the `alphashape` library defines the alpha parameter (α) such that the radius of the

curving tool is computed as $1/\alpha$, which is the reverse of the definition provided in Chapter 2. Three values of the curving parameter were tested, namely 0.5, 1.0, and 1.5. The appropriate result was selected based on whether the resulting mesh contained no gaps and included all of the points, while at the same time fitting the points as tightly as possible. For most trees, a parameter value of 1.5 yielded the desired outcome. Lastly, all the individual alpha shape meshes were collected, incorporated in one OBJ and created the whole scene of the tree area. Regarding the assigned materials, for the February representation only the wood material was used, while for the June representation only the leaf material.(**Figure 4.25**). The workflow of this case is presented at **Figure 4.26**.

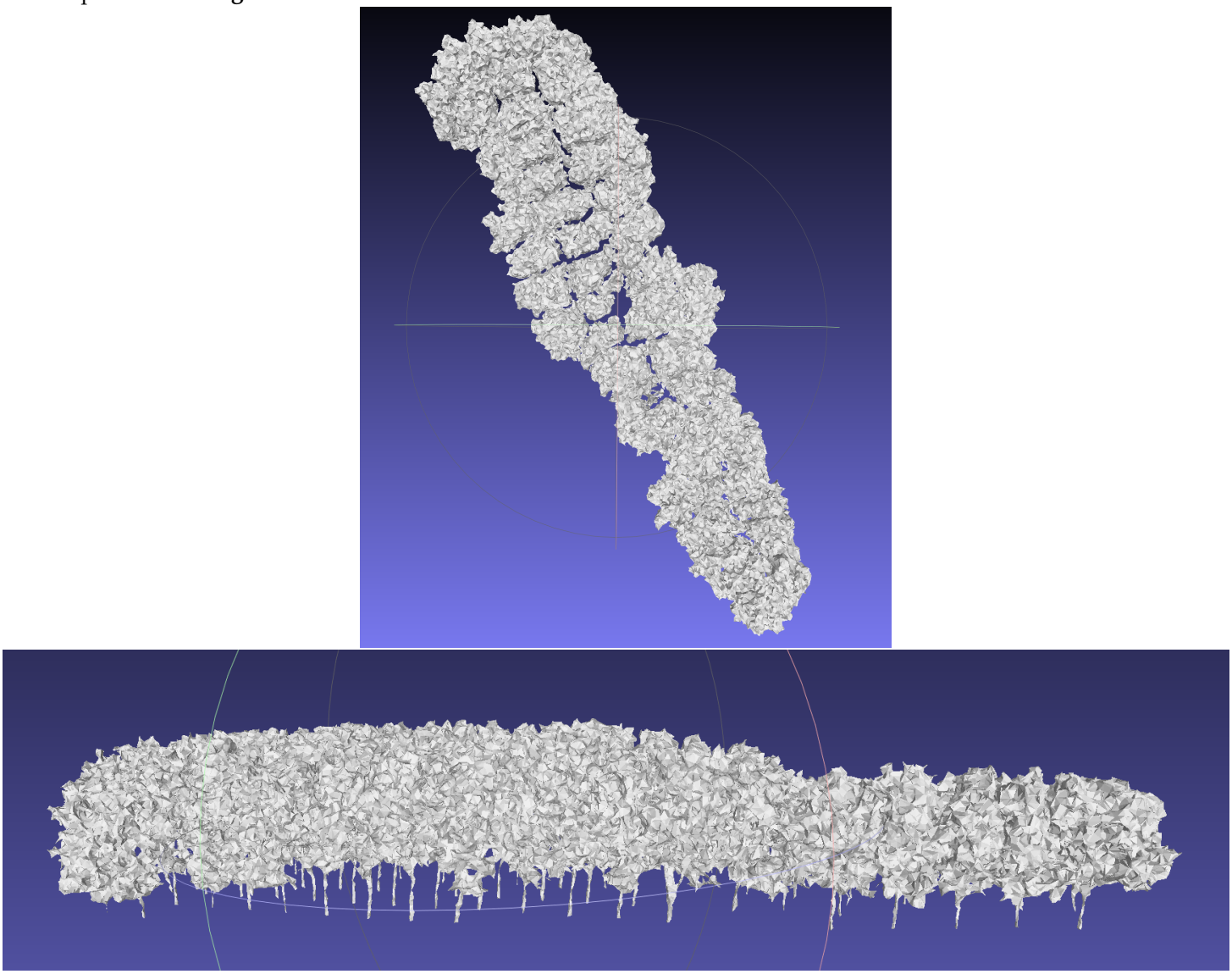


Figure 4.25.: February representation of the Alpha Shape Case from two perspectives.

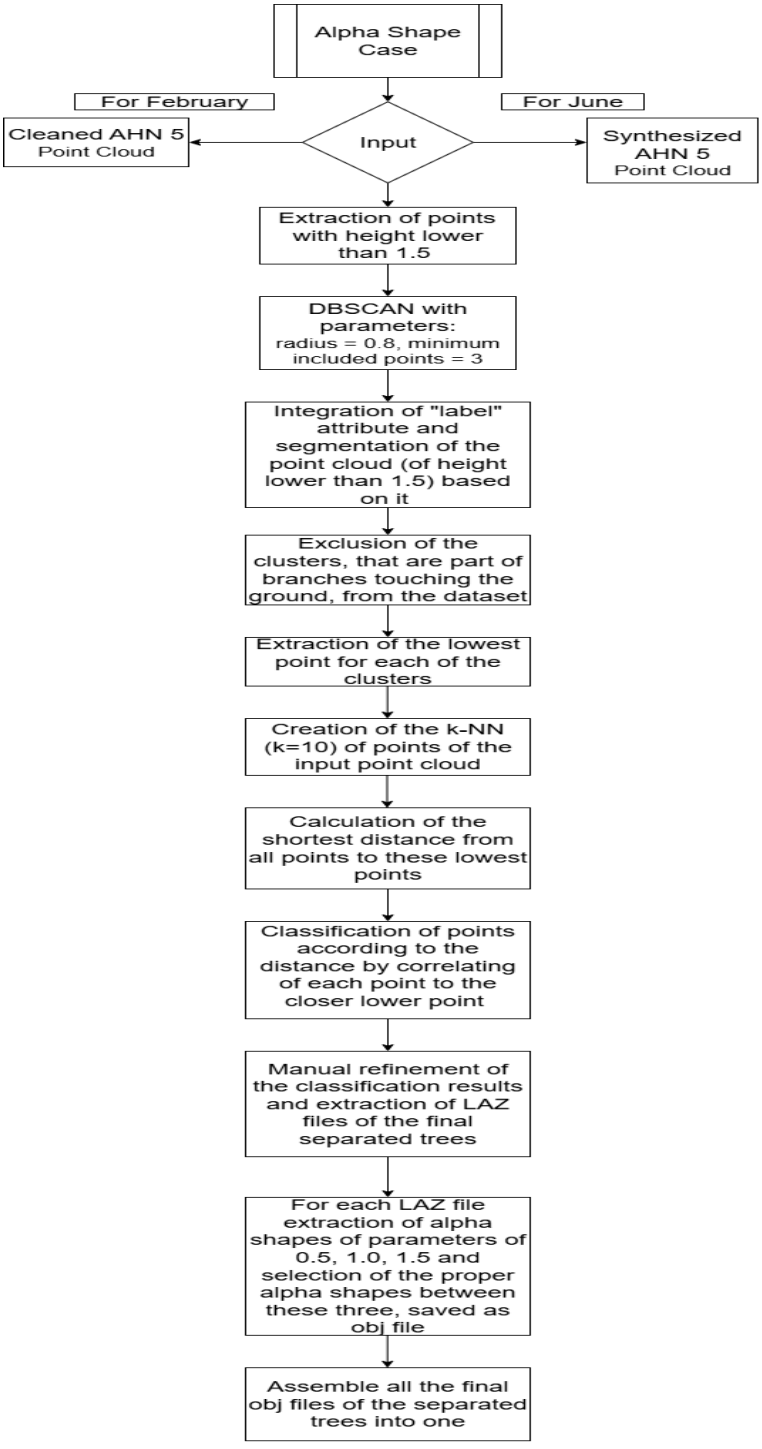


Figure 4.26.: Flow chart of the Alpha Shape Case.

4.4. Daylight Simulation

Based on what was mentioned earlier in this Chapter, for now out of the parameters needed for the performance of the Daylight Availability workflow, the location and solar data and the materials of all the tree representation have already determined. For the rest of the object of the scene, some assumptions were made, as illustrated at **Table 4.6**. These materials were utilized in every case and both in February and June to create the same conditions for the surrounding environment.

Table 4.6.: Material per object category

Object Category	Reflectance (%)
Ground	15.7
Building Parts (walls, roof, columns, floor)	43.24
Door	44.16
Glass on the Facade	15
Bridge	43.24

In addition, based on **Figure 1.2**, the occupied area was defined as a small rectangle surface at the position the West Sensor approximately is installed. Its normal vector was computed by designing a vertical to the surface vector from its center and then calculating the unit vector of the direction which points towards the tree area. The result normal vector of the occupied area is $\boldsymbol{n} = (-0.926, -0.376, 0)$. Then, on this surface four sensors were defined on Rhino configuration settings to later calculate and use their mean value per every time stamp so as to increase the accuracy of the resulting illuminance values.

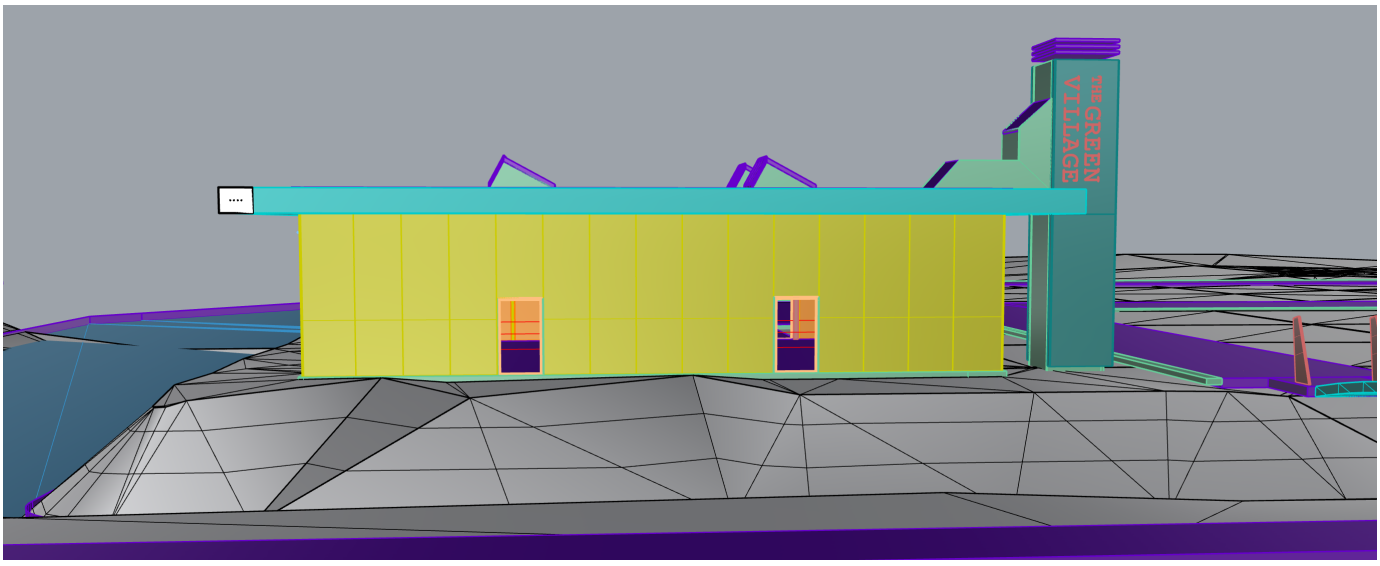


Figure 4.27.: Occupied surface on the west facade of the CCC Building.

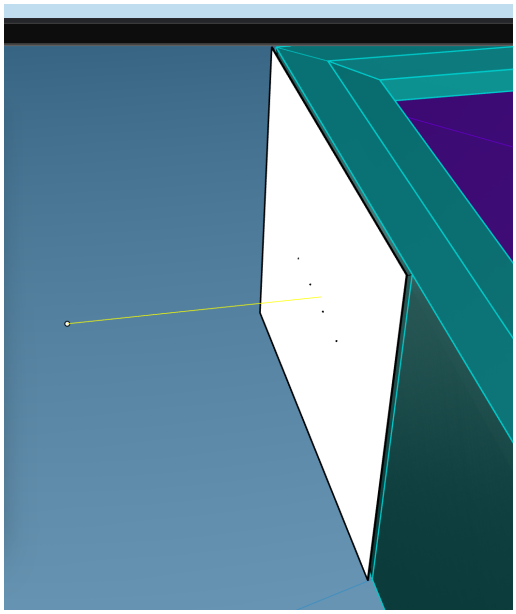


Figure 4.28.: Occupied surface with four sensors and the design aid used to compute the surface normal.

5. Results And Analysis

This section presents the analysis made on the simulated illuminance values. After completing all February and June simulations, statistical metrics such as the RMSE and the MBE were employed to highlight the differences among the cases and how the geometric representation of the trees influenced the performance of the simulations. Also, the Voxel Grid Case which was proven to be the most successful was discussed in more depth using diagrams of nearest voxel distance distributions and error distribution histograms and, as well as, bar charts of hourly MBE values for each voxel size and material demonstrating variations of the simulated values under clear sky, intermediate and cloudy sky conditions.

5.1. February results overview

RMSE per February Case		
Point Cloud Case		
7714.33		
Alpha Shape Case		
4352.70		
Voxel Grid Case		
Sizes (m)	1st approach	2nd approach
0.03	6698.80	7125.65
0.05	4575.95	4941.21
0.07	3998.52	4012.75
0.10	4127.83	3656.13
0.30	4469.17	4216.66
0.50	4502.08	4398.05
Convex Hull Case		
4562.66		

MBE per February Case		
Point Cloud Case		
-1982.30		
Alpha Shape Case		
1061.87		
Voxel Grid Case		
Sizes (m)	1st approach	2nd approach
0.03	-1628.73	-1808.02
0.05	-614.10	-899.17
0.07	109.00	-131.93
0.10	714.37	-149.34
0.30	1159.34	1049.28
0.50	1180.15	1173.23
Convex Hull Case		
1267.58		

Figure 5.1.: RMSE and MBE values in lux for February simulations.

For the February simulations, the Point Cloud - based Case resulted not only in the highest RMSE value of 7714.33 lux, but also in the strongest overestimation of illuminance values, as indicated by the highest negative MBE (-1982.30 lux). The Alpha Shape Case performed better compared to the Point Cloud Case. However, its MBE of 1061.87 lux indicates that the simulated illuminance values were considerably lower than expected. The Convex Hull Case yielded an RMSE of 4562.66 lux, which was lower than the Point Cloud Case,

but its higher positive MBE (1267.58 lux) reduced its suitability as a representation of this tree geometry for winter period.
To first compare these three cases, the fact that they include different kind of geometry, but the same wood material was utilized for them should be taken into consideration. With this opaque material remaining the same across all cases, the Point Cloud - based Case represents the trees through sparse small cubes, yet the other two cases provide broader and more solid 3D representations. In other words, regarding the Point Cloud - based Case the density of the original AHN 5 point cloud is not adequate to illustrate the trees, but at the same time the group of meshes at the Alpha Shape Case and the massive object representing the convex hull of the branch points from all the trees collectively are not appropriate to portray the trees accurately in this time period. Therefore, there is a strong deviation between the sensor's records and these simulated illuminance values for these cases. Also, given the contrasting results arising from the geometric differences among the aforementioned cases in terms of overestimation and underestimation of the illuminance values, it can be concluded that a representation which balances object sparsity (capturing branch porosity) while ensuring consistent 3D form would be more suitable for the February daylight simulations.
Focusing on the Voxel Grid Case, results showed a strong dependency both on voxel size for both of the approaches. The voxel size of 0.03 m resulted in high RMSE values (6698.80 and 7125.65 lux for the first and second approach, respectively) and a pronounced overestimation of illuminance, as indicated by the strongly negative MBE values of -1628.73 and -1808.02 lux for the first and second approach, respectively.
As the voxel size increased, the RMSE and MBE improved producing a RMSE of 3998.52 lux and a MBE of just 109.0 lux at size 0.07 m for the first approach, whereas a RMSE of 3656.13 lux and a MBE of -149.34 lux at size 0.10 m for the second approach. A worth mentioning simulation with good results as well was performed following the second approach and using size of 0.07 m and generated a RMSE of 4012.75 lux and a MBE of -131.93 lux. However, the improvement of the RMSE and MBE values stops at these sizes. For the next two sizes, there is a gradually rise of both RMSE and MBE values, yet they do not exceed the levels of deviation observed at 0.03 m. In general, voxel grid sizes between 0.07 m and 0.10 m produce results that align more closely with the real sensor measurements throughout the entire time series.

5.1.1. Whole Time Series Analysis — Voxel Grid Case

Regarding the first approach, **Figure 5.1** shows that the RMSE values decreased gradually from 0.03 to 0.07 m, after which they increased steadily up to 0.50 m. Looking at the MBE values, this fluctuation manifests differently: at 0.03 m there was a strong overestimation of the illuminance values, which decreased rapidly at 0.05 m. From 0.07 to 0.50 m the MBE values rose steadily, indicating that 0.07 m yielded the lowest level of underestimation of the examined metric.
Since the the same wood material was only used in this approach, the differences in the results may stem from variations in voxel sparsity or consistency. The following diagrams (**Figures 5.2–5.7**) show that, as voxel size increased, the distances of each voxel to its nearest voxel were reduced. More specifically, at sizes 0.03 and 0.05 m, most voxels have nearest voxel distances in the range 0.055–0.083 m with a mean of 0.109 m (**Figure 5.2**) and 0.054–0.081 m with a mean of 0.092 m (**Figure 5.3**), respectively, suggesting, as the corresponding diagrams also show, that many voxels had nearest voxel distances outside these ranges. In contrast, for sizes 0.30 and 0.50 m, the majority of nearest voxel distances are between 0.000 and 0.018 m, with mean values of 0.008 m (**Figure 5.6**) and 0.002 m (**Figure 5.7**), respectively. For sizes 0.07 and 0.10 m, intermediate behavior was observed. For the former, nearest voxel distances are mostly 0.027–0.054 m with mean value of 0.076 m (**Figure 5.4**), while for the latter this range was 0.000 to 0.026 m with mean value of 0.056 m (**Figure 5.5**).
From these statistic metrics, it is apparent that the voxels at sizes 0.03 and 0.05 m have the largest distances from each other, as they remain more closely tied to the points of the dataset. It can be said that these two representations resemble the Point Cloud - based Case, which did not sufficiently block the light generated

in the simulation. On the other hand, the sizes of 0.3 and 0.5 m produced more compact representations and in conjunction with Rhino material (which reflects 35.9 % of the incident light), significantly hindered the passage of the light beams from the target surface to the external environment. Consequently, the sizes of 0.07 and 0.10 m yielded more moderate and balanced representations, maintaining a favorable trade-off between porosity and solidity of the involving 3D objects.

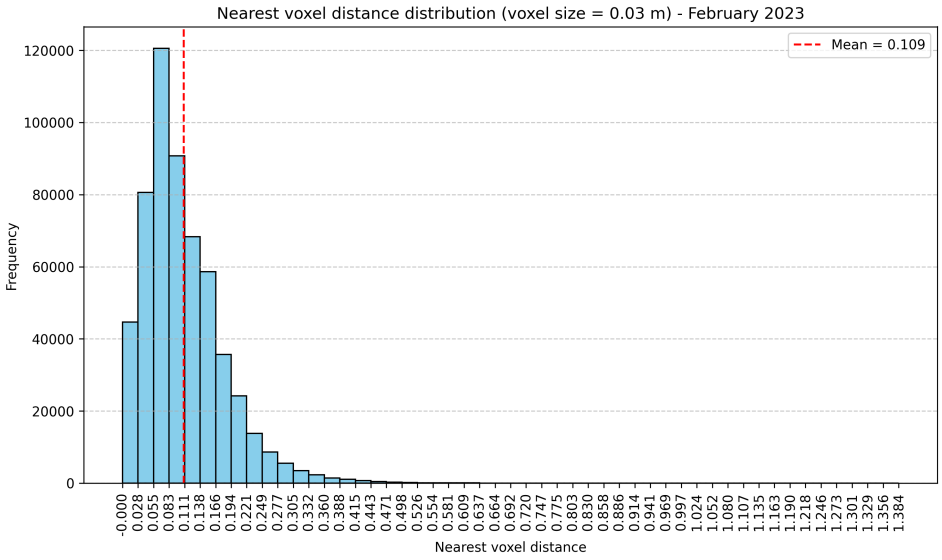


Figure 5.2.: Nearest voxel distance distribution at size 0.03 m in February.

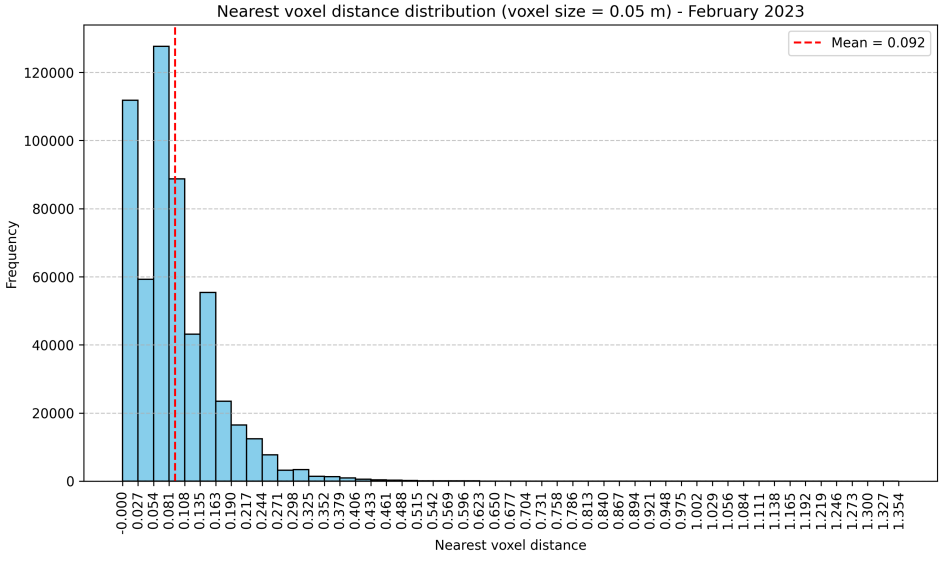


Figure 5.3.: Nearest voxel distance distribution at size 0.05 m in February.

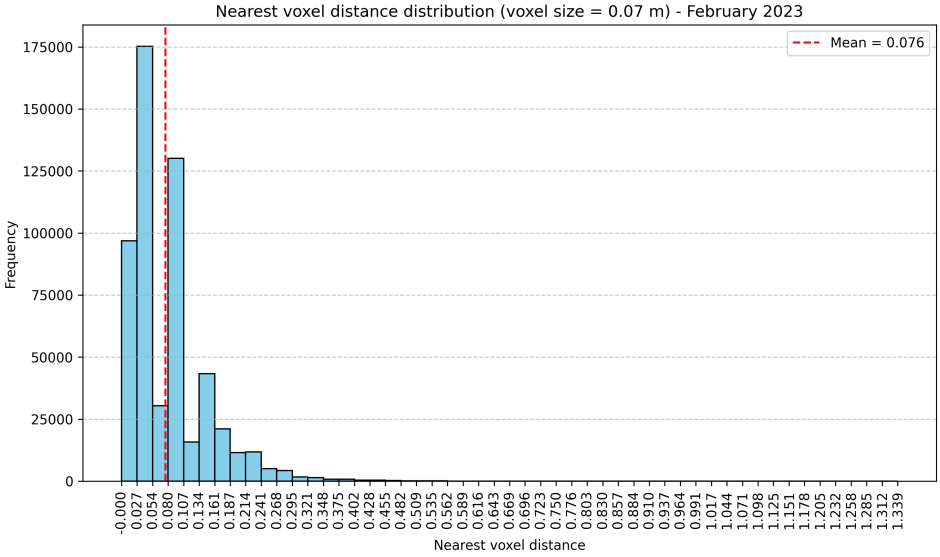


Figure 5.4.: Nearest voxel distance distribution at size 0.07 m in February.

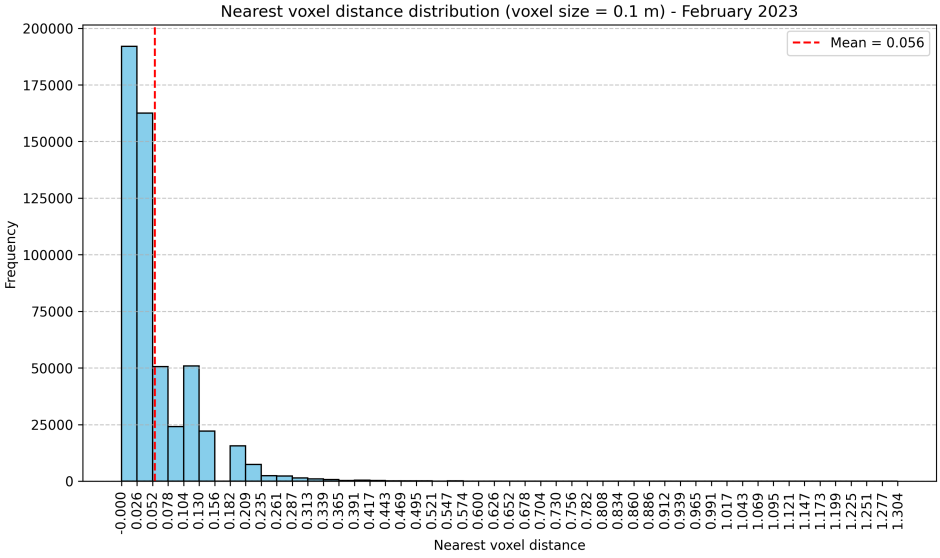


Figure 5.5.: Nearest voxel distance distribution at size 0.10 m in February.

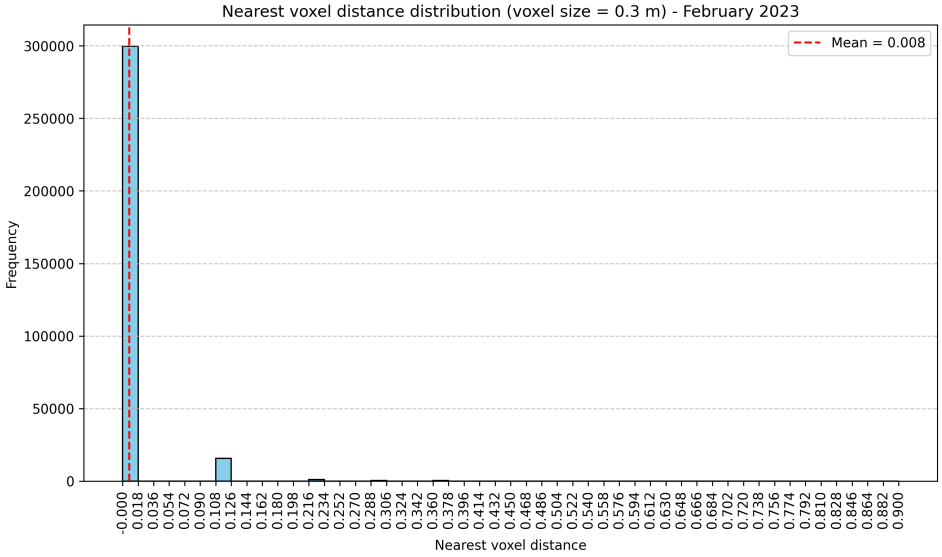


Figure 5.6.: Nearest voxel distance distribution at size 0.30 m in February.

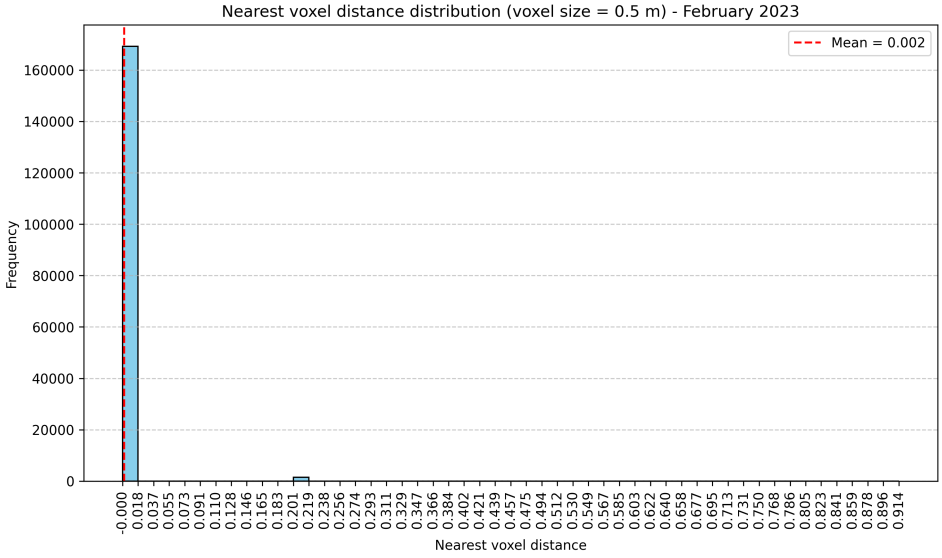


Figure 5.7.: Nearest voxel distance distribution at size 0.50 m in February.

Additionally, **Figures** 5.8–5.13 illustrate the behavior of errors across the voxel grid simulations of the first approach. At smaller voxel sizes (0.03–0.05 m) (**Figures** 5.8 and 5.9), the distributions are wider, with errors extending to large negative values. Next, at size 0.07 (**Figures** 5.10) the distribution is narrower and there are less extreme errors, most of the error values being concentrated around the mean value. However, the more the voxel size increases (**Figures** 5.11 to 5.13), the more the distribution extends to the large positive values, not maintaining the level of balance of size 0.07 m.

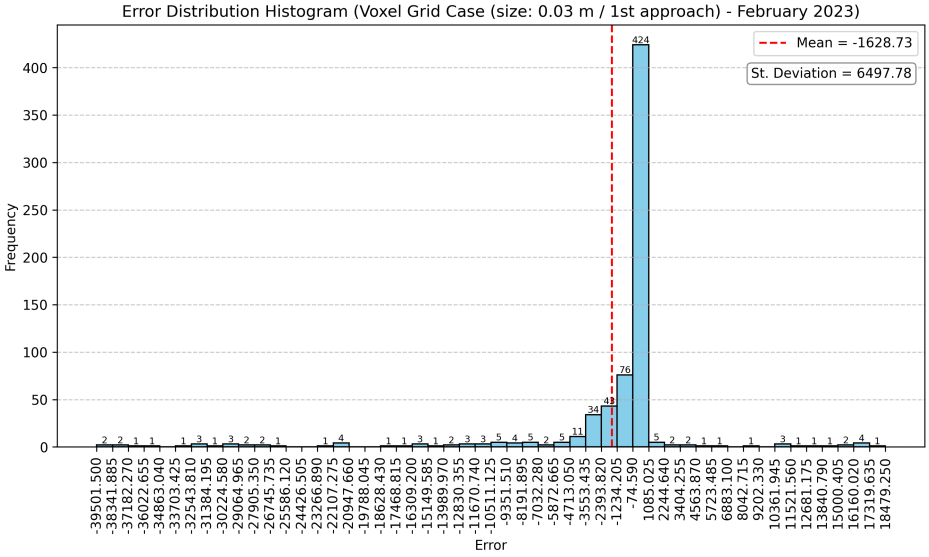


Figure 5.8.: Error distribution histogram at size 0.03 m based on the 1st approach in February.

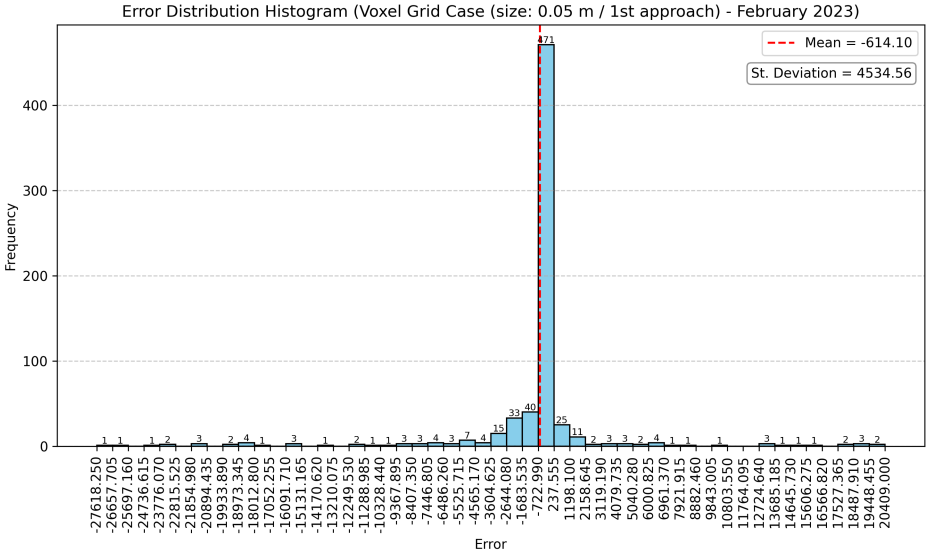


Figure 5.9.: Error distribution histogram at size 0.05 m based on the 1st approach in February.

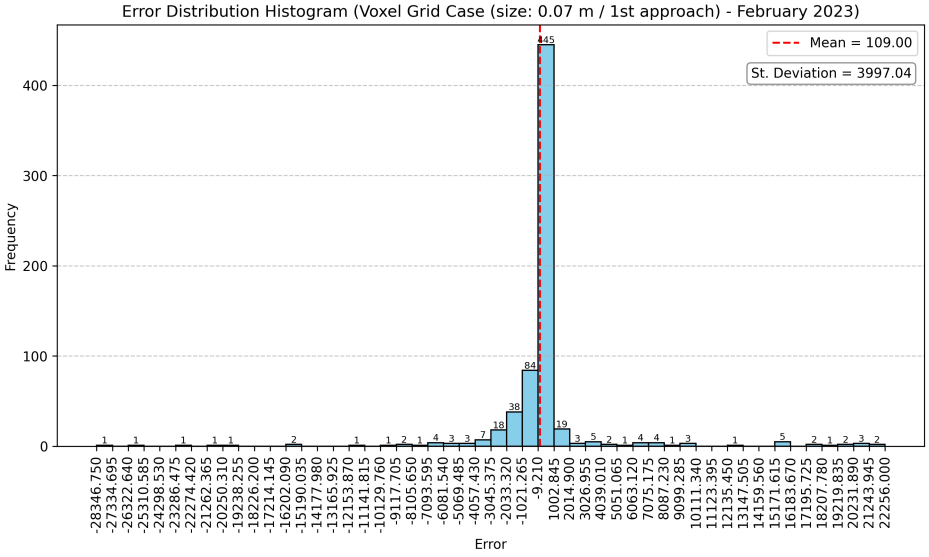


Figure 5.10.: Error distribution histogram at size 0.07 m based on the 1st approach in February.

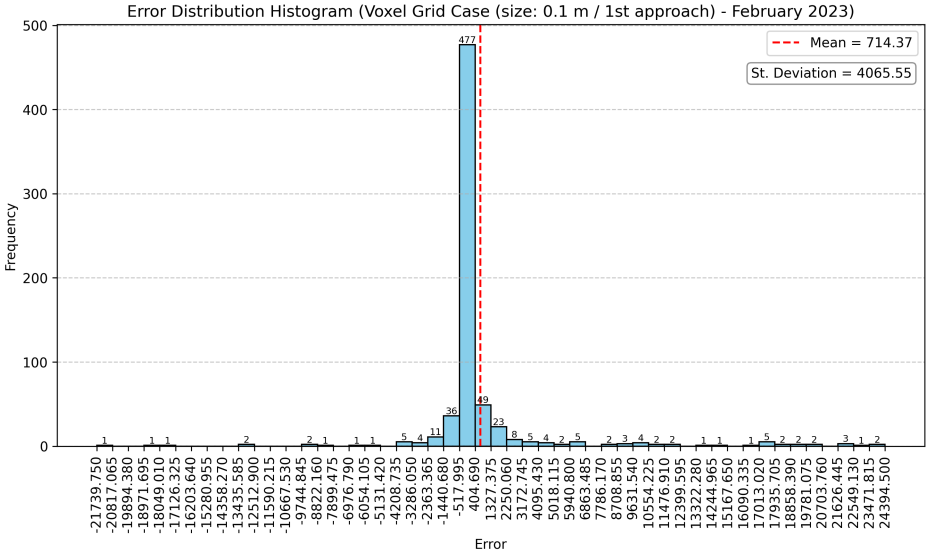


Figure 5.11.: Error distribution histogram at size 0.10 m based on the 1st approach in February.

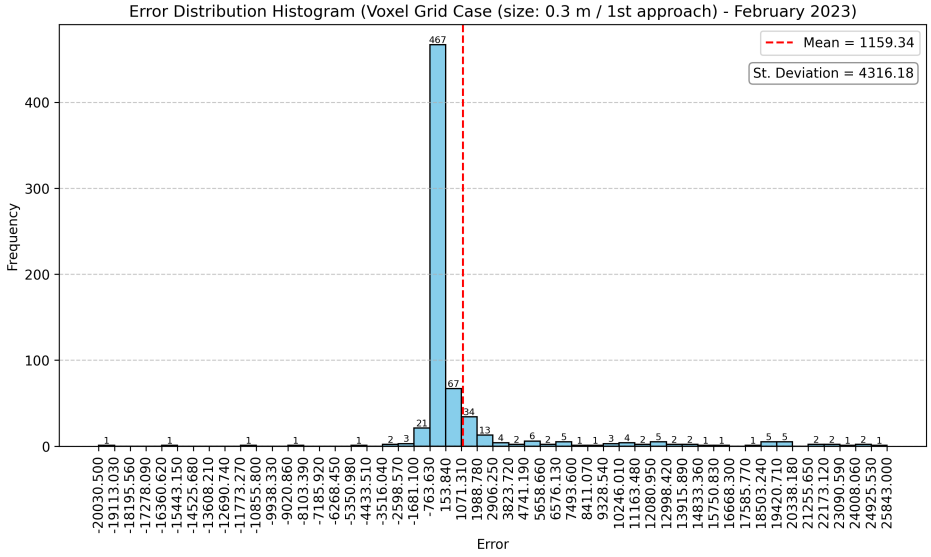


Figure 5.12.: Error distribution histogram at size 0.30 m based on the 1st approach in February.

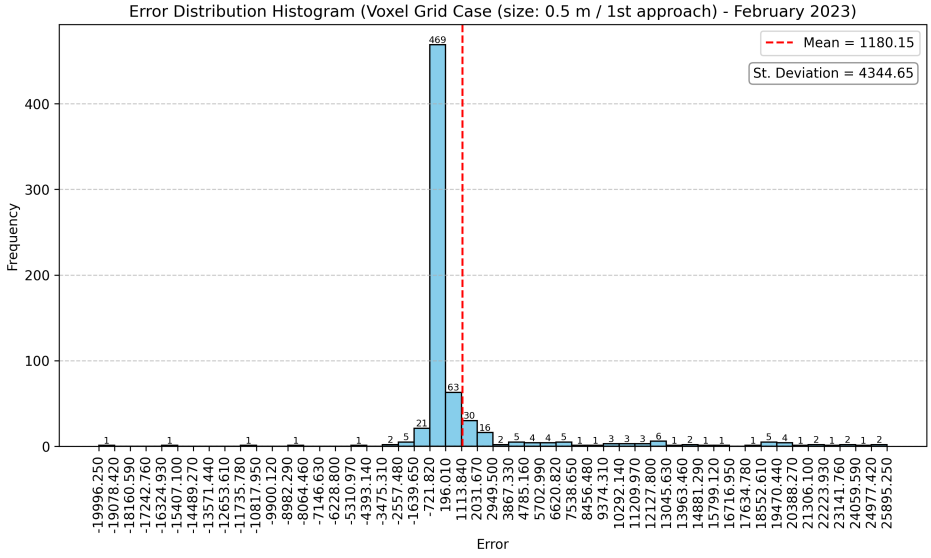


Figure 5.13.: Error distribution histogram at size 0.50 m based on the 1st approach in February.

With respect to the second approach, as explained earlier, the voxel sizes of 0.07 and 0.10 m proved to be the most effective, as they achieved improved RMSE and MBE values (**Figure 5.1**).

Table 5.1.: Voxel grid distributions by size and transparency class for February simulations.

Grids	0.03 m	0.05 m	0.07 m	0.10 m	0.30 m	0.50 m
1st (transmittance: 88 %)	0	0	0	505,979	252,691	127,057
2nd (transmittance: 63 %)	560,281	553,232	539,381	27,025	60,649	38,153
3rd (transmittance: 38 %)	1,311	4,918	11,767	1,353	3,595	5,240
4th (transmittance: 12.5 %)	1	41	225	49	64	235

According to **Figure 5.1, 5.2 and 5.3**, the poor performance of the first two lower voxel sizes can be attributed to the high sparsity of the voxels, since they do not sufficiently obstruct the incident light. However, based on **Table 5.1** the better results at size of 0.07 m can arise from the synergy between the right distribution of the different kinds of voxels in terms of their transmittance and of the more moderate distances to the nearest voxel. In particular, the larger number of the voxels of transmittance of 38% and 12.5% at 0.07 m compared to the sizes of 0.03 and 0.05 m indicates a lower level of transparency in this geometric representation, which in turn leads to less overestimation of the illuminance values.
In addition, among the other three sizes, the 0.10 m case provides the most appropriate balance between the various voxels. The first grid at 0.10 m contains significantly more voxels than the others and in conjunction with the given distances between the voxels in this case (**Figure 5.5**), the provided estimation of the illuminance values is more accurate. On the other hand, regarding the sizes of 0.30 m and 0.50 m, the substantially higher number of the third and fourth grid combined with the low inter-voxel distances (**Figures 5.6** and **5.7**), contribute to illuminance values that are much lower than expected.
A notable observation regarding sizes 0.07 m and 0.10 m through the two different approaches is the kind of their resulting estimation. Based on **Figure 5.1**, while the first approach results in an underestimation of the illuminance values, the second approach give rise to an overestimation. Interestingly, the level of deviation among these sizes in the second approach does not change significantly, in contrast to the first approach, where the underestimation is more pronounced at 0.10 m than at 0.07 m. The variation in the nature of the results stems mainly from the different materials assigned to the voxels in each approach. On the one hand, the first approach involves only an opaque material, so light transmittance is negligible to non-existent, resulting in limited illuminance at the scene's target. On the other hand, in the second approach glass materials of various transmittance indices yields overestimation.
As illustrated by the error distribution histograms of the second approach (**Figures 5.14–5.19**), the same behavior observed in the corresponding histograms of the first approach is evident. The error distributions for sizes of 0.07 m and 0.10 m are more symmetric and centered closer to zero, indicating minimal bias, even though the illuminance values themselves are relatively large. On the contrary, it is apparent that at 0.03 and 0.05 m the errors are skewed toward large negative values, whereas at 0.30 m and 0.50 m many large positive errors appear, affecting the balance of the distribution.

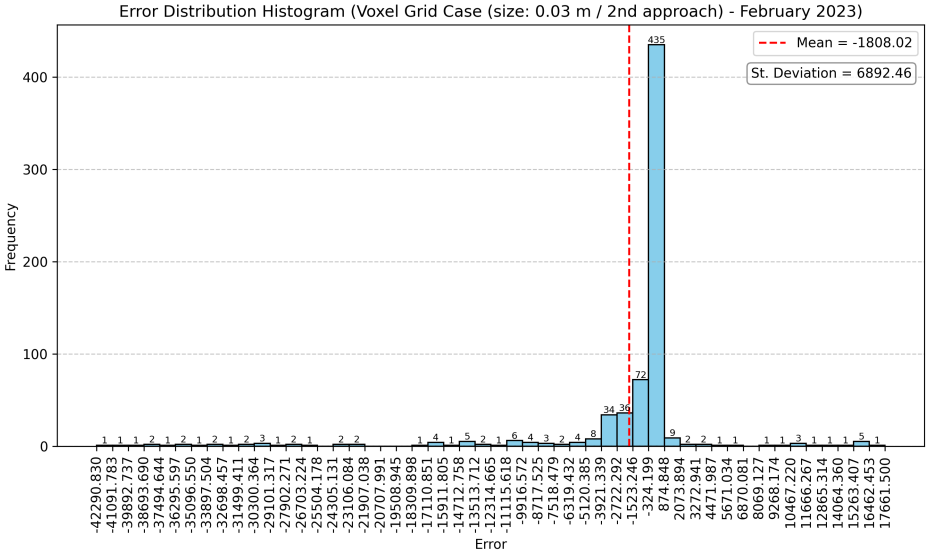


Figure 5.14.: Error distribution histogram at size 0.03 m based on the 2nd approach in February.

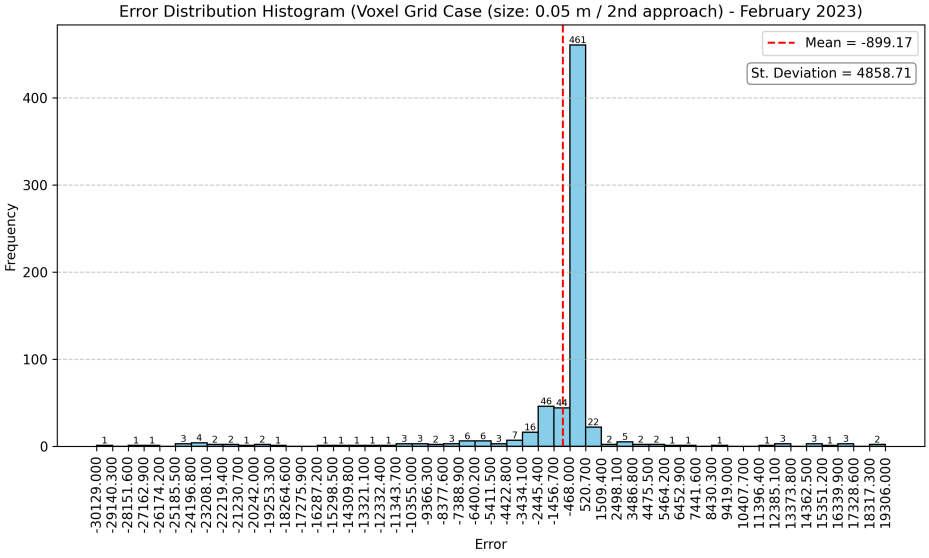


Figure 5.15.: Error distribution histogram at size 0.05 m based on the 2nd approach in February.

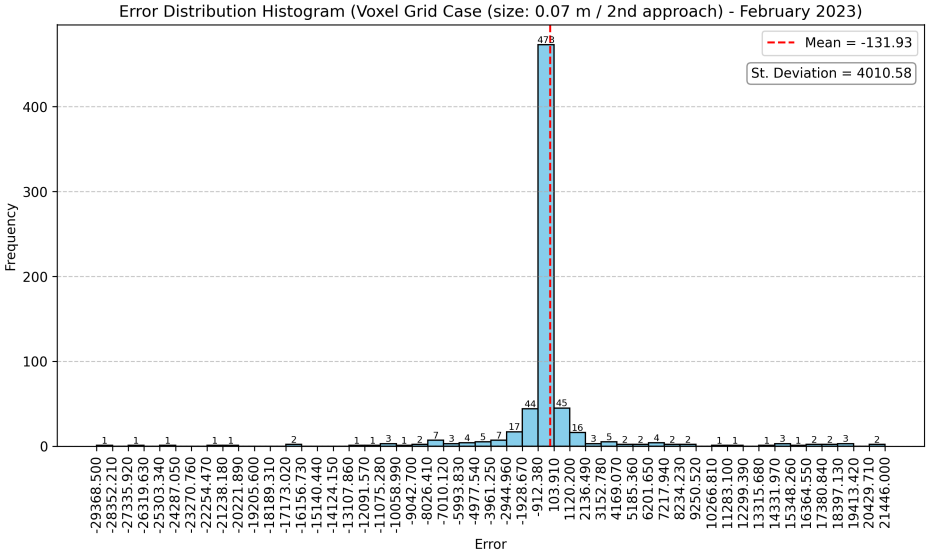


Figure 5.16.: Error distribution histogram at size 0.07 m based on the 2nd approach in February.

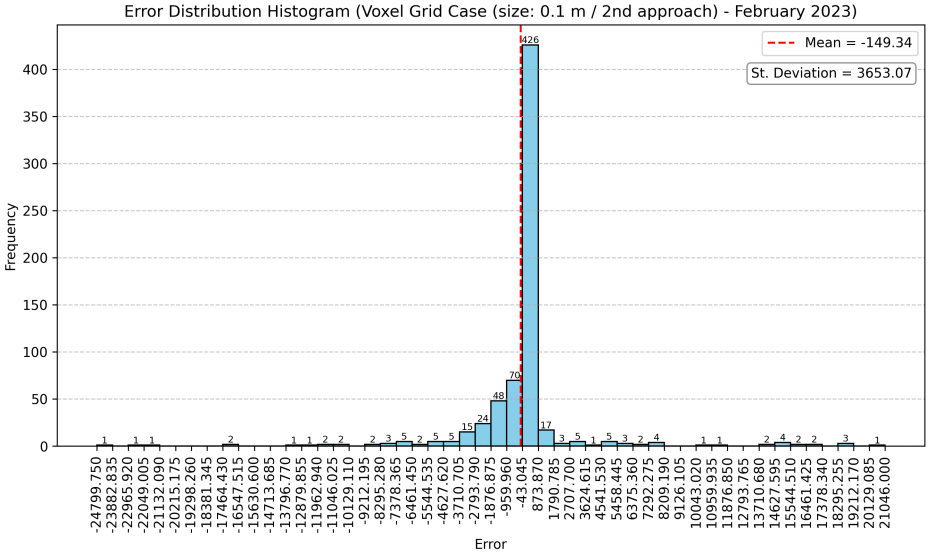


Figure 5.17.: Error distribution histogram at size 0.10 m based on the 2nd approach in February.

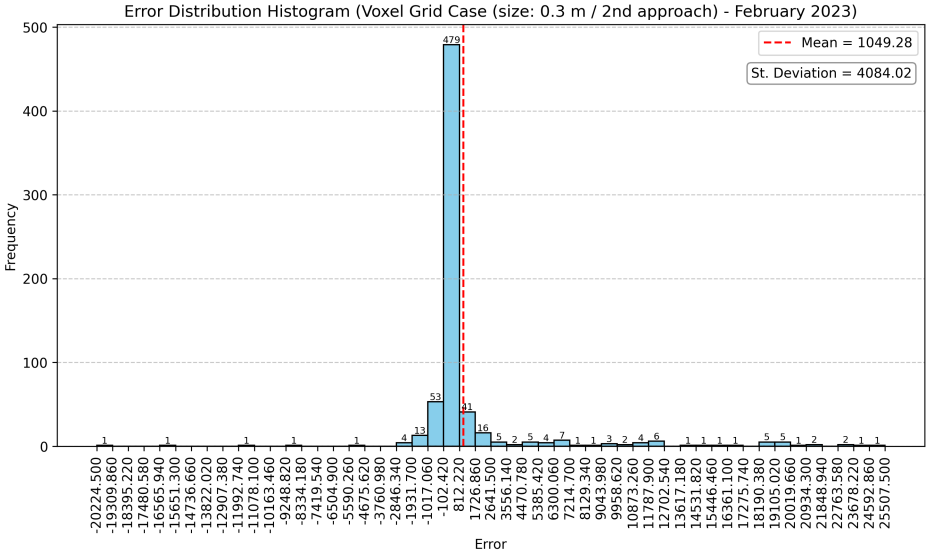


Figure 5.18.: Error distribution histogram at size 0.30 m based on the 2nd approach in February.

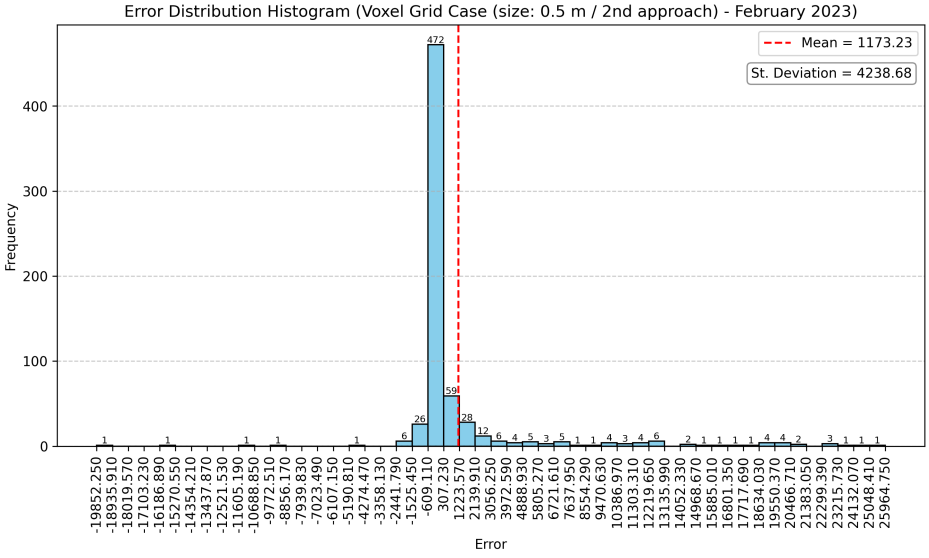


Figure 5.19.: Error distribution histogram at size 0.50 m based on the 2nd approach in February.

5.1.2. Weather Condition Analysis — Voxel Grid Case

The results were also assessed regarding of the Voxel Grid Case simulations responded in different weather conditions. Regarding the selection of the representative days, clear sky days exhibit profiles with high maximum illuminance values and steep rise and fall slopes, indicating predominantly direct sunlight and minimal cloud obstruction throughout the day. In contrast, intermediate days have lower peaks and slightly flattened or irregular shapes, something that is consistent with partial cloudiness. Lastly, cloudy days demonstrate low curves, indicating diffuse light with overcast conditions. As mentioned in Chapter 3, Grafana platform was used to distinguish better which days should be selected. For February as shown in Figure 5.20, the five selected days for each weather condition are: (a) 7/2, 8/2, 14/2, 26/2, and 28/2 for clear sky conditions, (b) from 19 to 23/2 for cloudy conditions, and (c) 5/2, 6/2, 15/2, 16/2, and 18/2 for intermediate conditions.

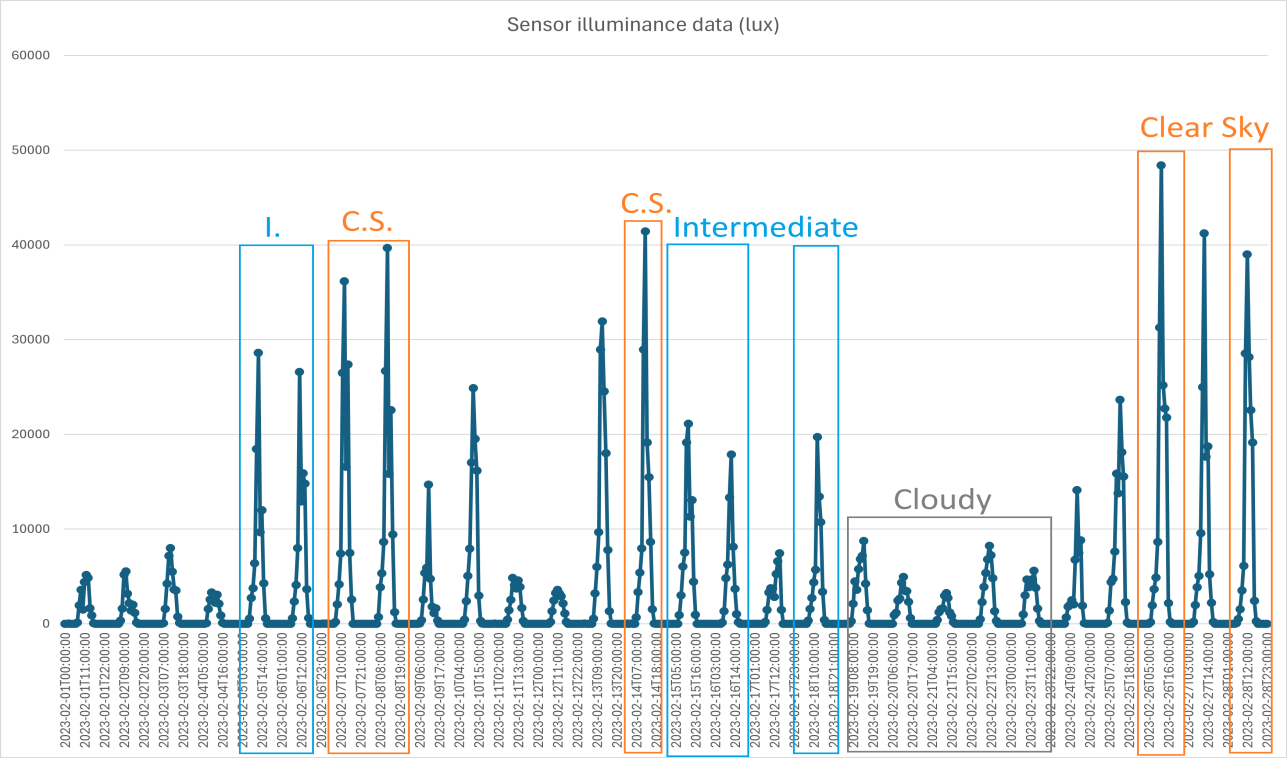


Figure 5.20.: Representative days for each sky condition in February.

Based on Figures 5.21 and 5.22, noticeable deviations between the sensor and simulated illuminance values can be observed on clear sky days, particularly pronounced between 11:00 and 16:00 UTC (12:00 and 17:00 local time), regardless of the voxel material. More specifically, at 11:00 and 12:00 UTC (12:00 and 13:00 local time) the illuminance value is underestimated in all voxel cases, while from 13:00 to 16:00 UTC (14:00 and 17:00 local time) the behavior changes depending on the voxel size and material. As confirmed from *Sunrise and sunset times* website [62], the solar noon time for February in the Netherlands is around 12:55 local time. The higher illuminance sensor values at 11:00 and 12:00 UTC (12:00 and 13:00 local time) are attributed to

the high solar altitude during these hours, which translates to more direct solar radiation to reach the Earth's surface. At these altitudes, the sunlight passes through a thinner atmospheric layer, resulting in reduced scattering, reflection, and absorption by air molecules and aerosols, and consequently higher measured illuminance. Therefore, inaccuracies in voxel geometry, material properties of the scene objects, the solar position derived from the sky model, or the georeferencing of the CCC building can result in pronounced deviations, as direct sunlight is highly intense, spatially localized and closely linked to the in-situ illuminance measurements. For the hours between 13:00 and 16:00 UTC (14:00 and 17:00 local time), the deviations will be further examined later in this chapter.

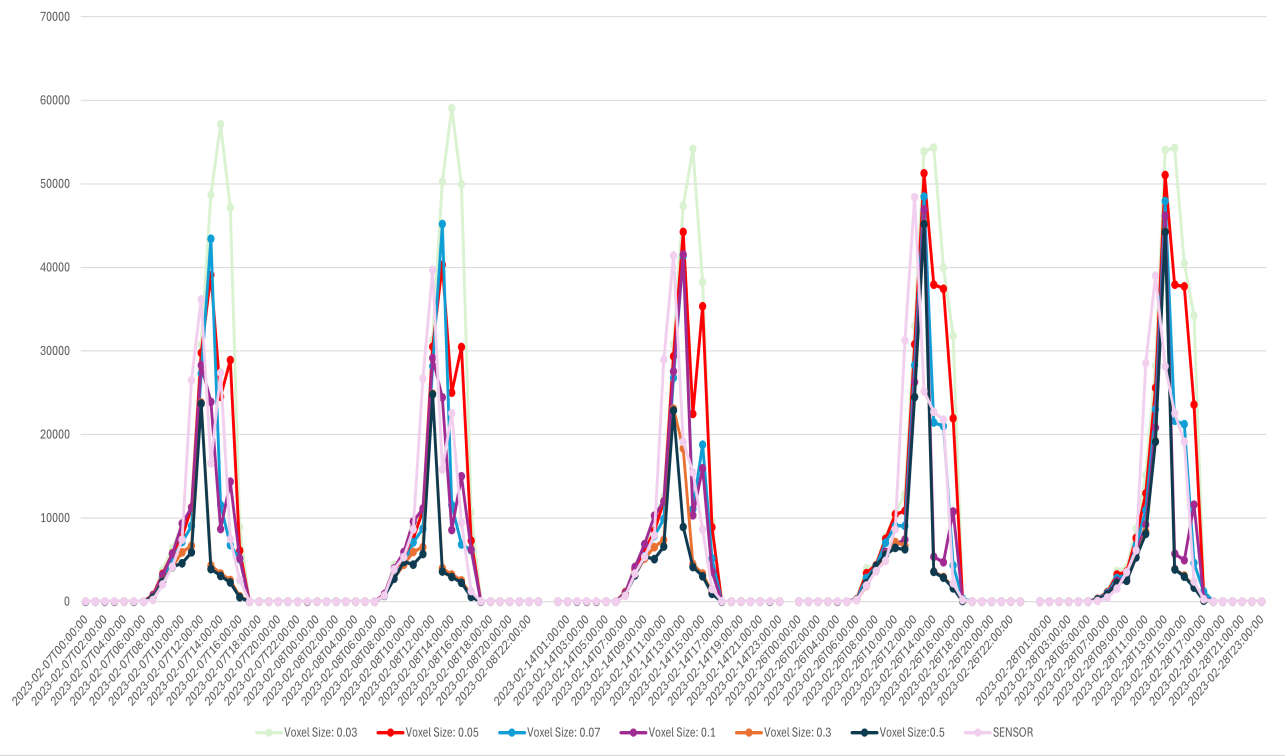


Figure 5.21.: Hourly illuminance values in lux from sensor and simulation for each voxel size using transparent voxels on five representative days under clear sky conditions in February.

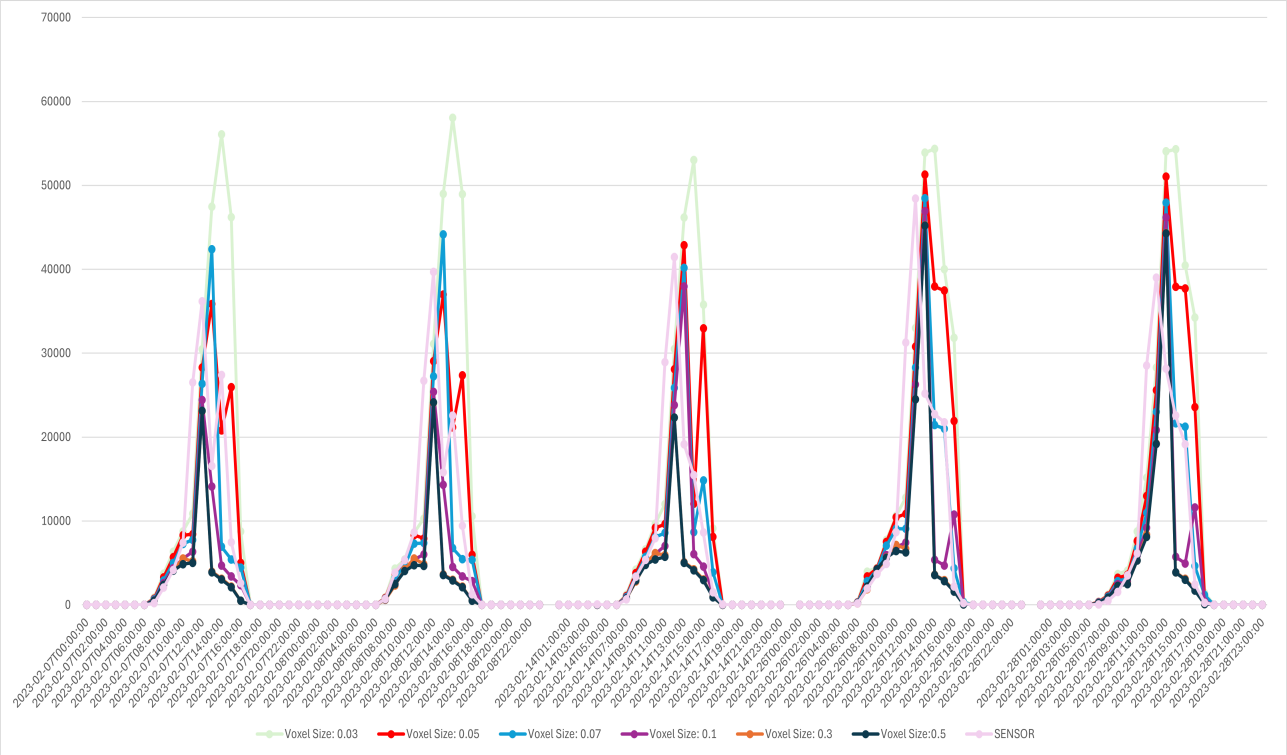


Figure 5.22.: Hourly illuminance values in lux from sensor and simulation for each voxel size using opaque voxels on five representative days under clear sky conditions in February.

With regard to cloudy days, the illuminance is dominated by diffuse light, meaning that radiation originates more from the entire sky dome rather than from a single solar position. As a result, the variations in illuminance between the sensor and the simulations are smaller. Additionally, from **Figures** 5.23 to 5.24 it can be observed that the simulated values most closely matching the sensor measurements correspond to voxel sizes of 0.3 m and 0.5 m, and to some extent 0.07 m and 0.1 m, regardless of the voxel material. This likely occurs because these voxel sizes and their spatial configuration provide a more stable and representative estimation of illuminance under diffuse lighting conditions.

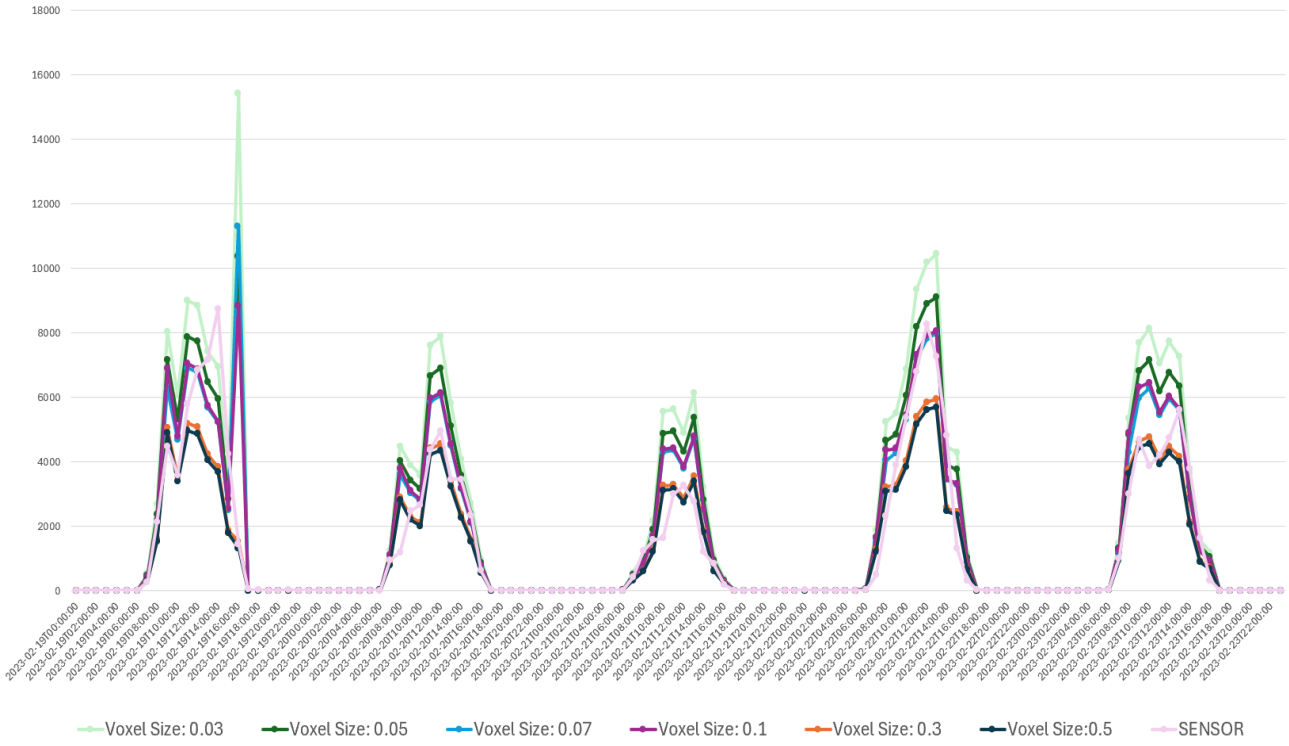


Figure 5.23.: Hourly illuminance values in lux from sensor and simulation for each voxel size using transparent voxels on five representative days under cloudy conditions in February.

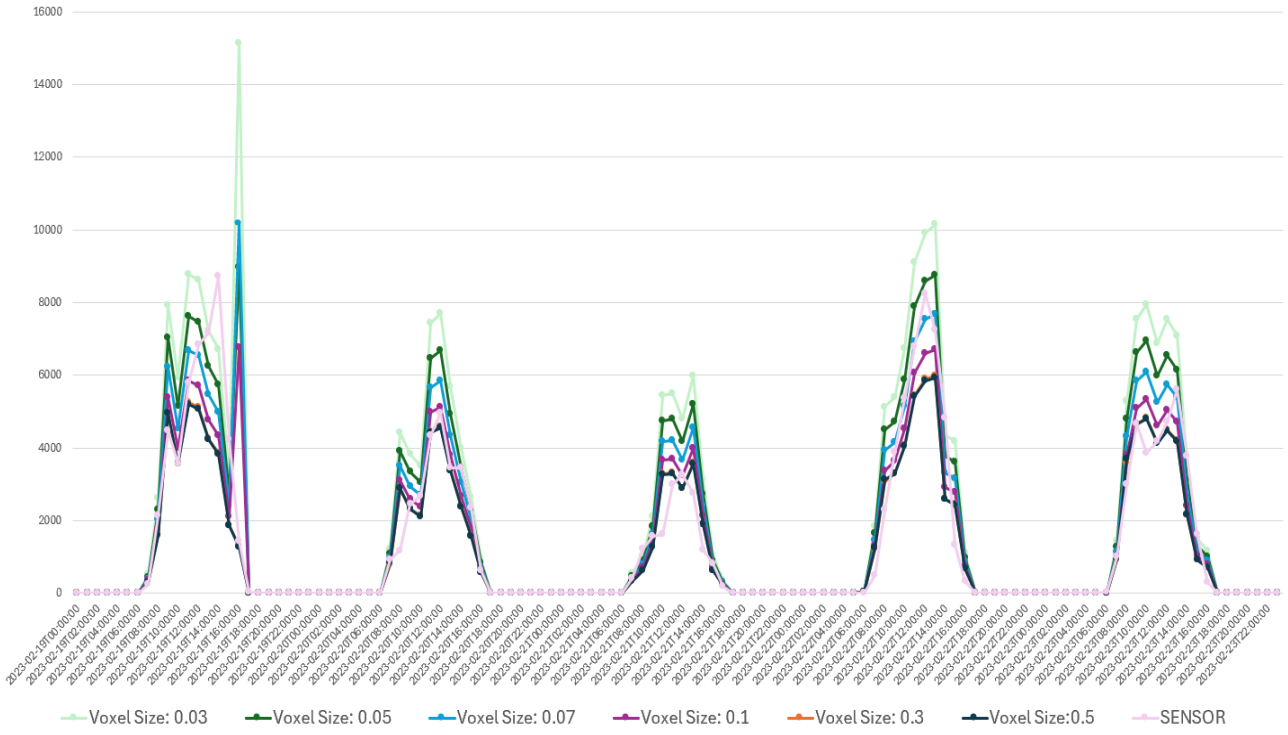


Figure 5.24.: Hourly illuminance values in lux from sensor and simulation for each voxel size using opaque voxels on five representative days under cloudy conditions in February.

For days with intermediate conditions (**Figures** 5.25 and 5.26), the differences between the sensor and the simulations are moderate, mainly due to the variability in cloud cover throughout these days. This is evident from 5/2 and 6/2, when the simulation results deviate more from the sensor measurements, as these days were sunnier compared to the others. In contrast to cloudy days, there is no clearly dominant voxel size that aligns best with the sensor data. However, the 0.03 m voxel case consistently shows lower accuracy than the other configurations, reflecting a similar pattern observed under all weather conditions.

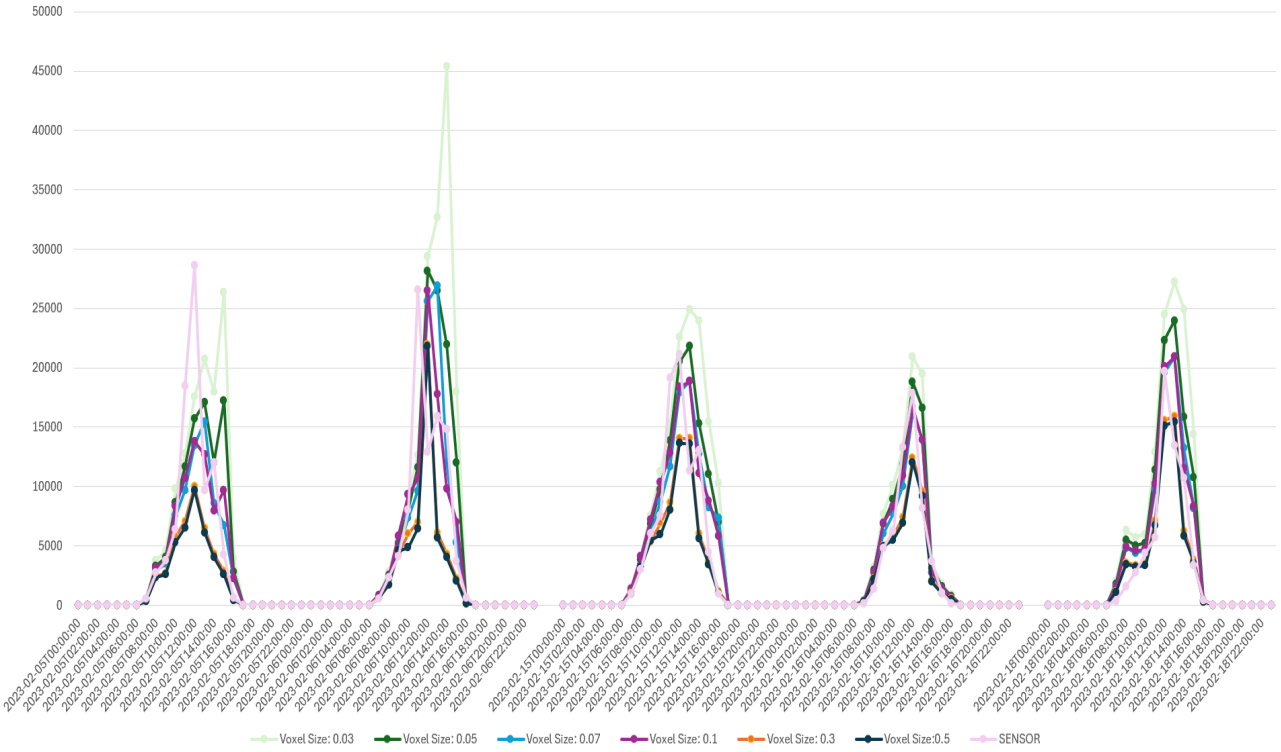


Figure 5.25.: Hourly illuminance values in lux from sensor and simulation for each voxel size using transparent voxels on five representative days under intermediate conditions in February.

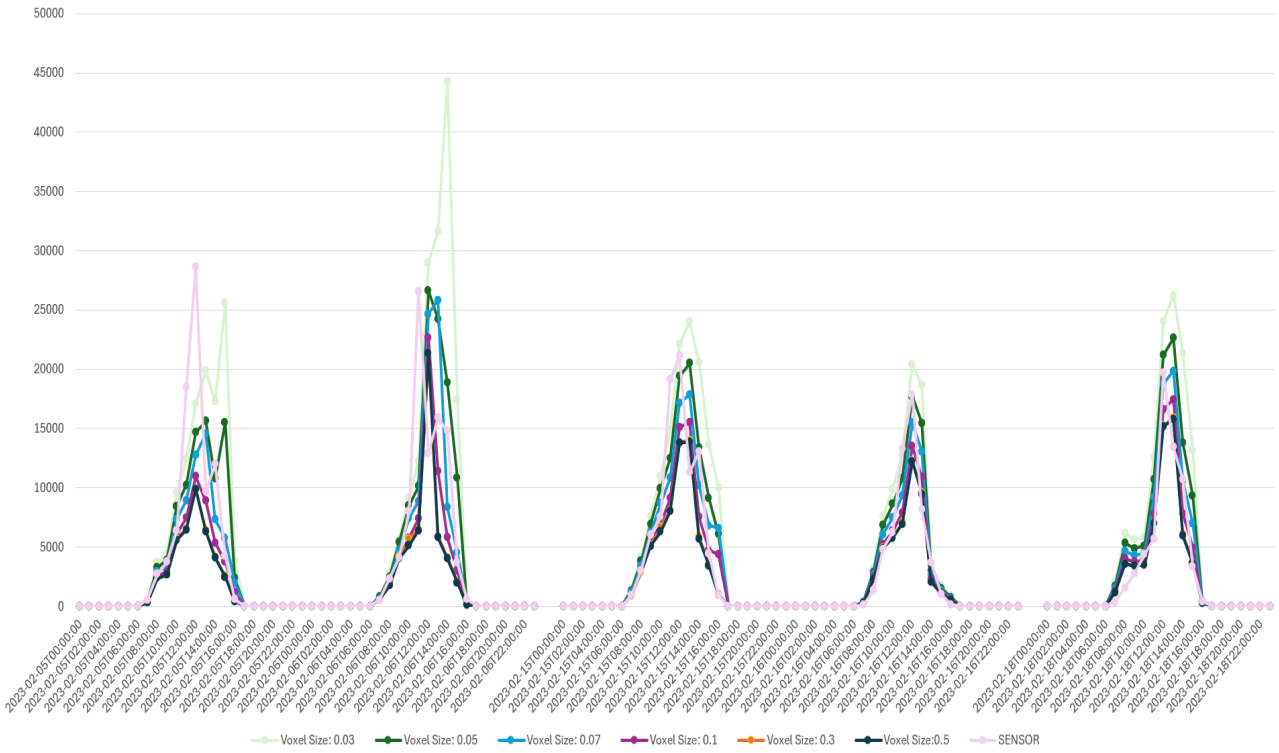


Figure 5.26.: Hourly illuminance values in lux from sensor and simulation for each voxel size using opaque voxels on five representative days under intermediate conditions in February.

From **Figure 5.27** to **5.38**, bar graphs of MBE values with their standard deviations for opaque and transparent voxels of all sizes in February were generated. The hours that were selected for these graphs are from 5:00 to 17:00 UTC, since the day in February lasts approximately from 6:30 to 16:30 UTC, so an hour before sunrise and an hour after sunset were examined as well. Generally, in this analysis, there are bigger errors, because the same differences in smaller samples have a more significant impact on the MBE calculation than in larger ones.

From **Figures 5.27, 5.28, 5.29**, and **5.30**, it can be observed that voxel grids of 0.03 m and 0.05 m, regardless of material, exhibit almost identical patterns per sky condition. From the beginning of the time series until 10:00 UTC (11:00 local time), the errors mainly indicate overestimation with low standard deviation, with those under clear sky conditions generally being smaller than those observed under cloudy or intermediate conditions.

Then, as has been shown from the previous diagrams, at 11:00 UTC there is a dramatic increase of the clear sky error, which also continues at 12:00 UTC (13:00 local time) at a slightly lower level. This can be attributed to the high solar altitudes that occur in these two hours, since they are close to the local solar noon (12:55 local time or 11:55 UTC). Interestingly, although the errors under intermediate sky conditions are lower than those under clear sky conditions, they exhibit larger standard deviations, reflecting the uncertainty introduced by the variability in cloud coverage. Lastly, from 13:00 to 16:00 UTC (14:00 to 17:00 local time), the clear sky simulated values are excessively overestimated, likely due to inaccuracies of simulation in calculating illuminance, combined with potential georeferencing inaccuracy of the CCC building and the small

voxel sizes. More specifically, since the sensor is oriented west and, as the sun moves toward the western horizon during these hours, the sensor receives a significant amount of direct sunlight, particularly around 13:00 UTC, when the solar altitude is still relatively high. Because the voxel size and spatial distribution are insufficient to accurately block the light, the simulations tend to overestimate the illuminance values. Regarding intermediate conditions, the errors follow the same pattern as under clear sky conditions but are lower in magnitude. In contrast, the errors under cloudy conditions are the smallest and remain negative, except at 16:00 UTC (17:00 local time), when they exceed those of the intermediate conditions. This anomaly is likely due to a sudden peak observed on 19/2 for voxel sizes between 0.03 and 0.1 m, probably caused by the wrongly simulated interaction between the voxel geometry and rays from the model at that time (**Figures 5.23** and **5.24**).
Regarding the noticeable variations between the 0.03 m and 0.05 m voxel cases, from 13:00 to 16:00 UTC the errors under clear sky conditions gradually decrease in magnitude for the 0.03 m case, while they fluctuate for the 0.05 m case. This indicates differences in how rays interact within the simulation, influenced by voxel geometry. Overall, errors from opaque voxels are higher than those from transparent ones during periods of strong underestimation (11:00–12:00 UTC, or 12:00-13:00 local time), whereas transparent voxels show higher errors during strong overestimation (13:00–16:00 UTC, or 14:00-17:00 local time), reflecting the stronger light attenuation caused by opaque voxels and the greater light permeability of transparent voxels.

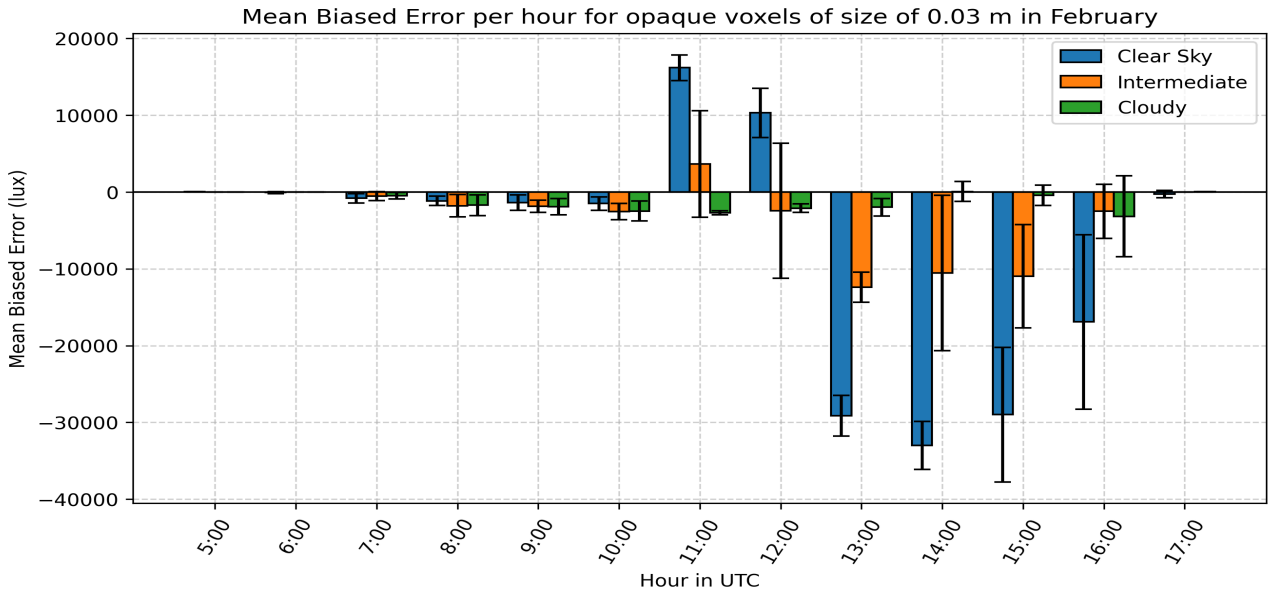


Figure 5.27.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.03 m in February under clear sky, cloudy, and intermediate conditions.

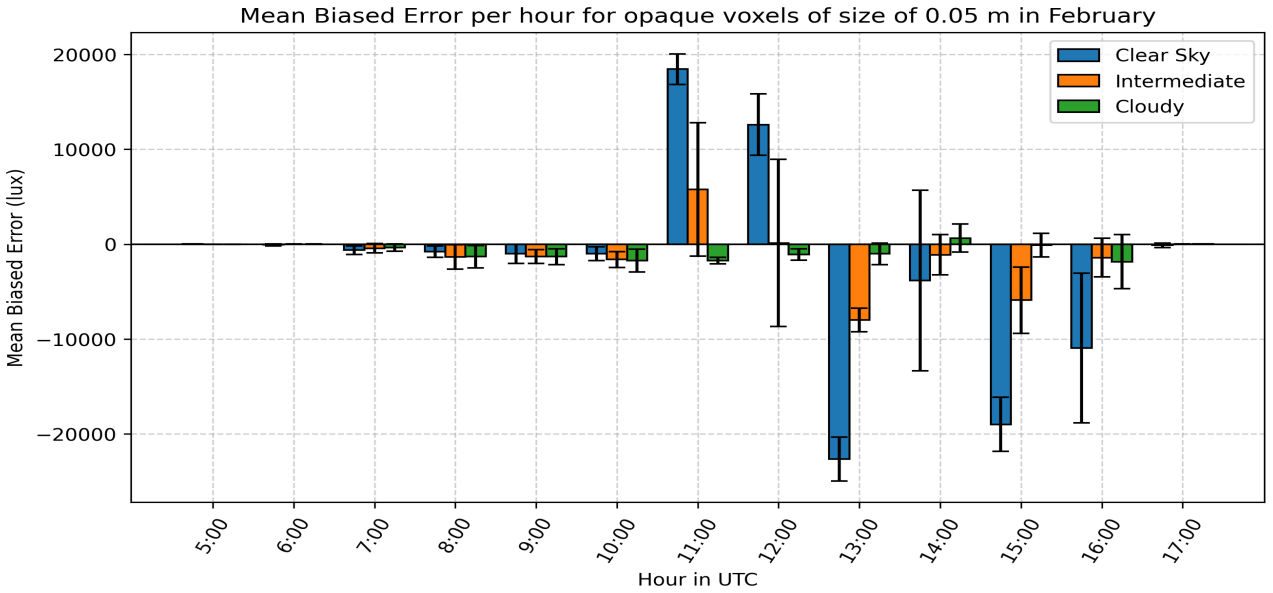


Figure 5.28.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.05 m in February under clear sky, cloudy, and intermediate conditions.

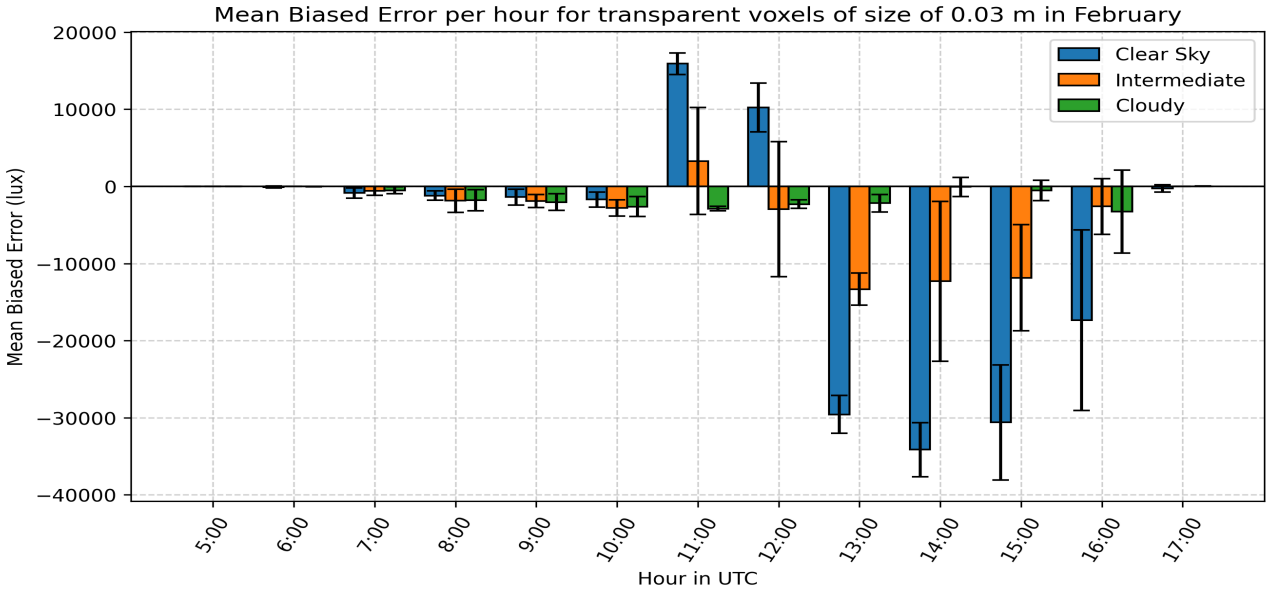


Figure 5.29.: Time series of Mean Biased Error with standard deviation per timestamp for transparent voxels of 0.03 m in February under clear sky, cloudy, and intermediate conditions.

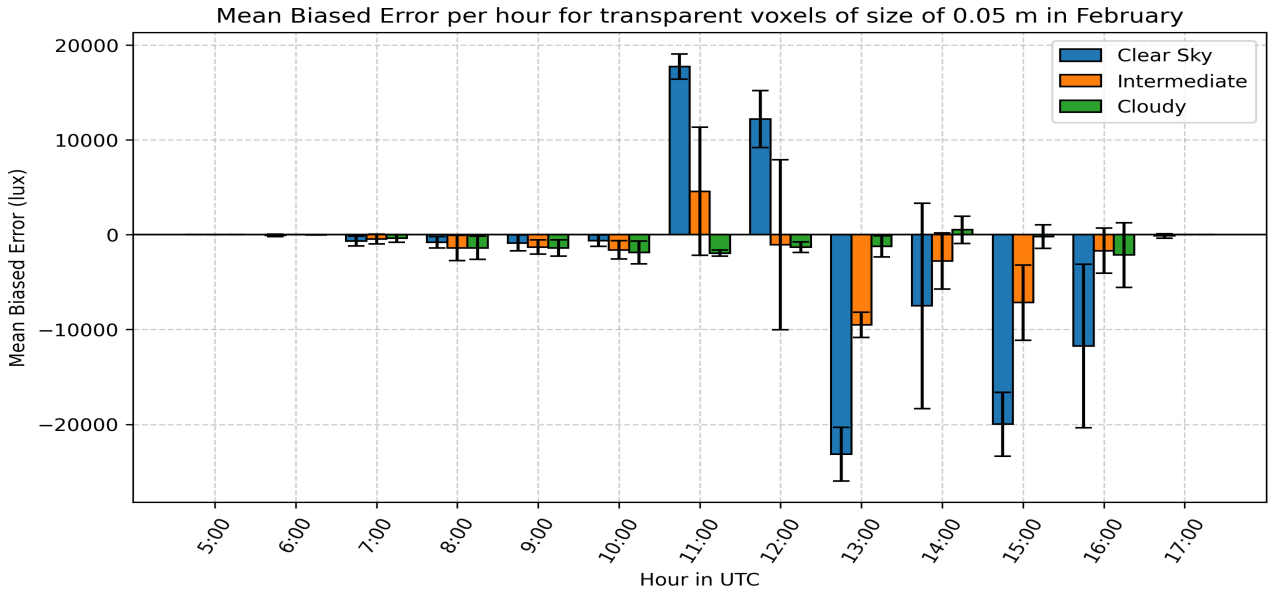


Figure 5.30.: Time series of Mean Biased Error with standard deviation per timestamp for transparent voxels of 0.05 m in February under clear sky, cloudy, and intermediate conditions.

Processing to voxels of sizes 0.07 m and 0.1 m for both materials, as shown in **Figures 5.31, 5.32, 5.33** and **5.34**, reveals that from the start of the time series until 13:00 UTC (14:00 local time), the behavior closely resembles that of the two smaller voxel sizes. In terms of magnitude, the errors remain generally smaller across all sky conditions until 10:00 UTC (11:00 local time), whereas between 11:00 and 13:00 UTC (12:00 and 14:00 local time), they become significantly larger compared to the previous voxel sizes, except for the cloudy condition dataset. This outcome, particularly under clear sky conditions, is associated with the increase in voxel size, as this causes the underestimation of the simulation to worsen, moving it further away from the sensor data. Subsequently, at 14:00 UTC (15:00 local time) the simulated illuminance values are underestimated in contrast of what happens at the sizes of 0.03 and 0.05 m and the smallest errors were achieved with the transparent voxels. For the next two hours, there is mainly overestimation of the illuminance values, except for clear sky conditions at 15:00 (16:00 local time) for size of 0.1 m, yet at transparent voxels of 0.07 m it is more moderate.

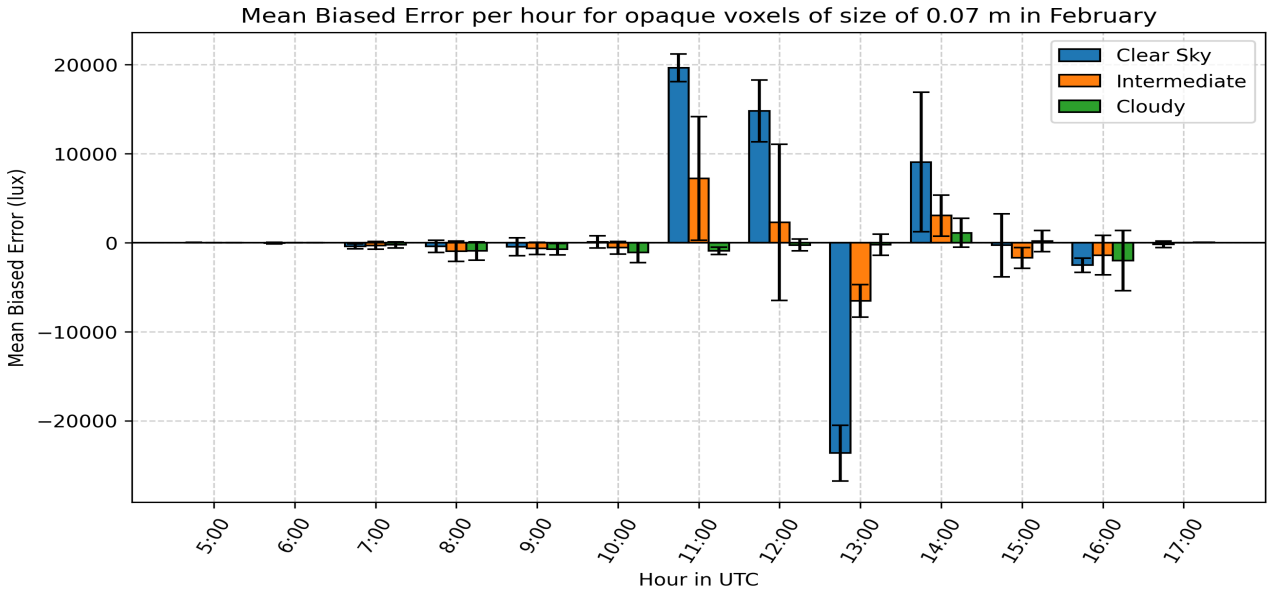


Figure 5.31.: Time series of Mean Biased Error with standard deviation per timestamp for opaque voxels of 0.07 m in February under clear sky, cloudy, and intermediate conditions.

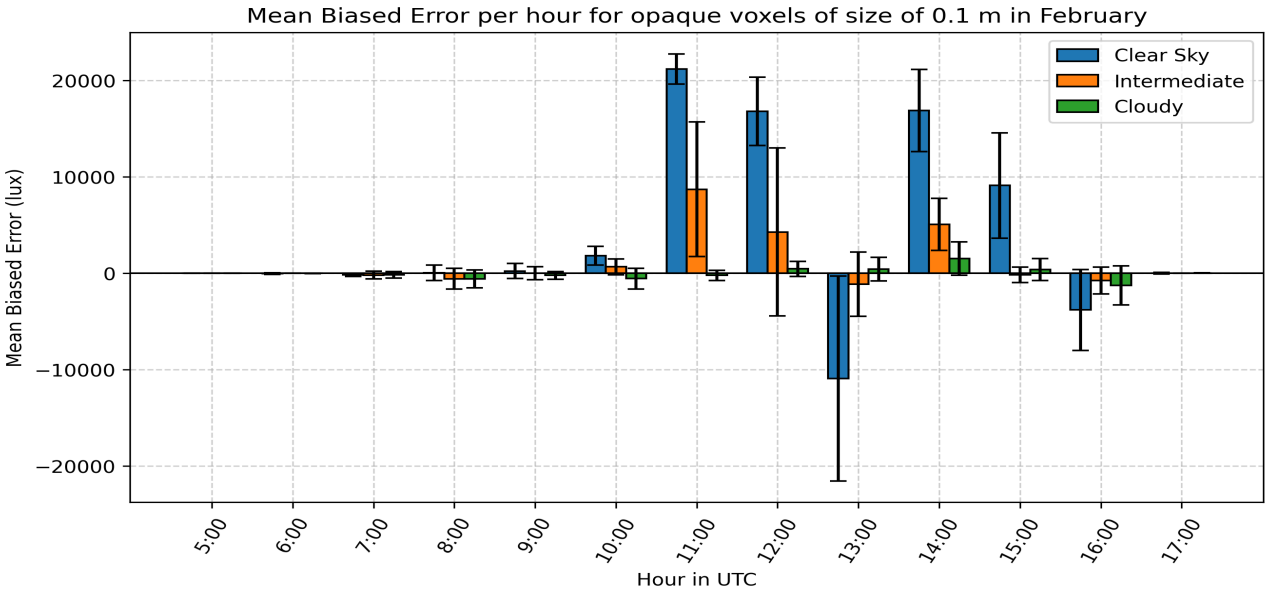


Figure 5.32.: Time series of Mean Biased Error with standard deviation per timestamp for opaque voxels of 0.1 m in February under clear sky, cloudy, and intermediate conditions.

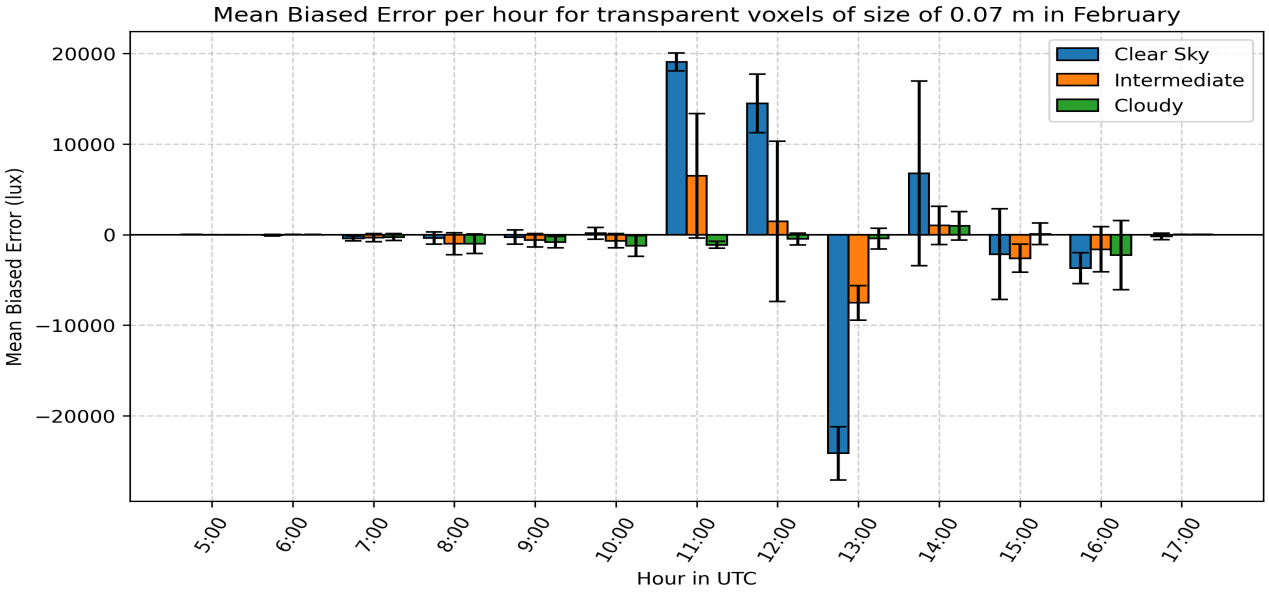


Figure 5.33.: Time series of Mean Biased Error with standard deviation per timestamp for transparent voxels of 0.07 m in February under clear sky, cloudy, and intermediate conditions.

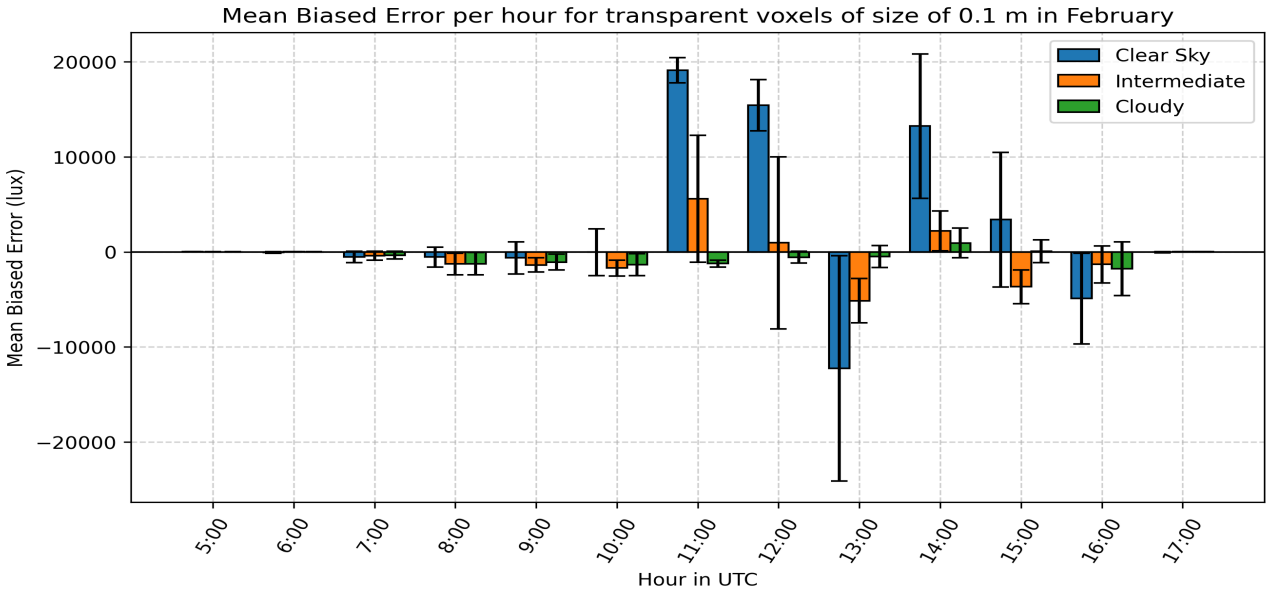


Figure 5.34.: Time series of Mean Biased Error with standard deviation per timestamp for transparent voxels of 0.1 m in February under clear sky, cloudy, and intermediate conditions.

Regarding voxel sizes of 0.3 and 0.5 m and the two material voxels (**Figures 5.35, 5.36, 5.37** and **5.38**), it is apparent that the rise of voxel size influenced the simulation results. At 11:00 and 12:00 UTC (12:00 and 13:00 local time) the underestimation got even higher, especially for clear sky and intermediate conditions, since the tree 3D representation that was created obscured the light significantly. The same happens with hours at 14:00 and 15:00 UTC, when there is still sunlight coming to the sensor because of its west orientation, yet the actual radiative behavior is not captured accurately due to the big size of the voxels. On the other hand, with these voxels the errors at 13:00 UTC (14:00 local time) are at all time low, but with extremely high standard deviations for clear sky and intermediate conditions, reveal variations in the sample of illuminance values. Lastly, it can be noticed that at this level of sizes, materials did not affect the simulation result considerably, apart from error at 13:00 UTC. However, for cloudy conditions, the cases of these sizes achieved the lowest errors and standard deviations. Yet, from 11:00 to 14:00 UTC (12:00 to 15:00 local time) , significant positive errors occurred due to underestimation, as even in cloudy conditions the simulated values did not approach those measured by the illuminance sensors.

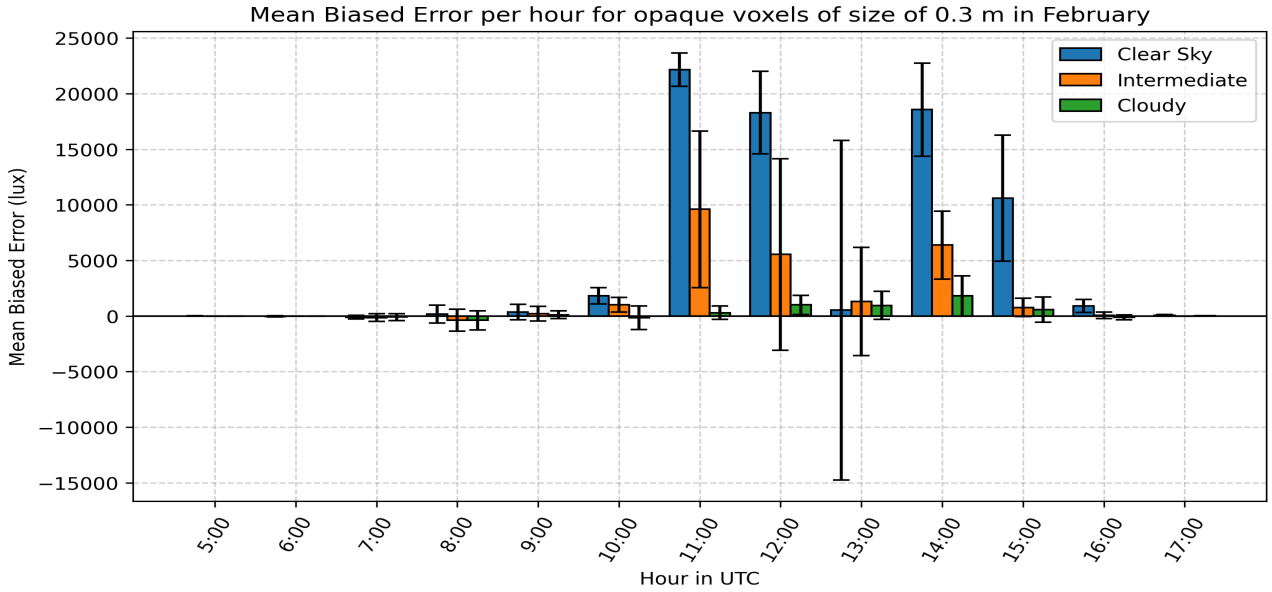


Figure 5.35.: Time series of Mean Biased Error with standard deviation per timestamp for opaque voxels of 0.3 m in February under clear sky, cloudy, and intermediate conditions.

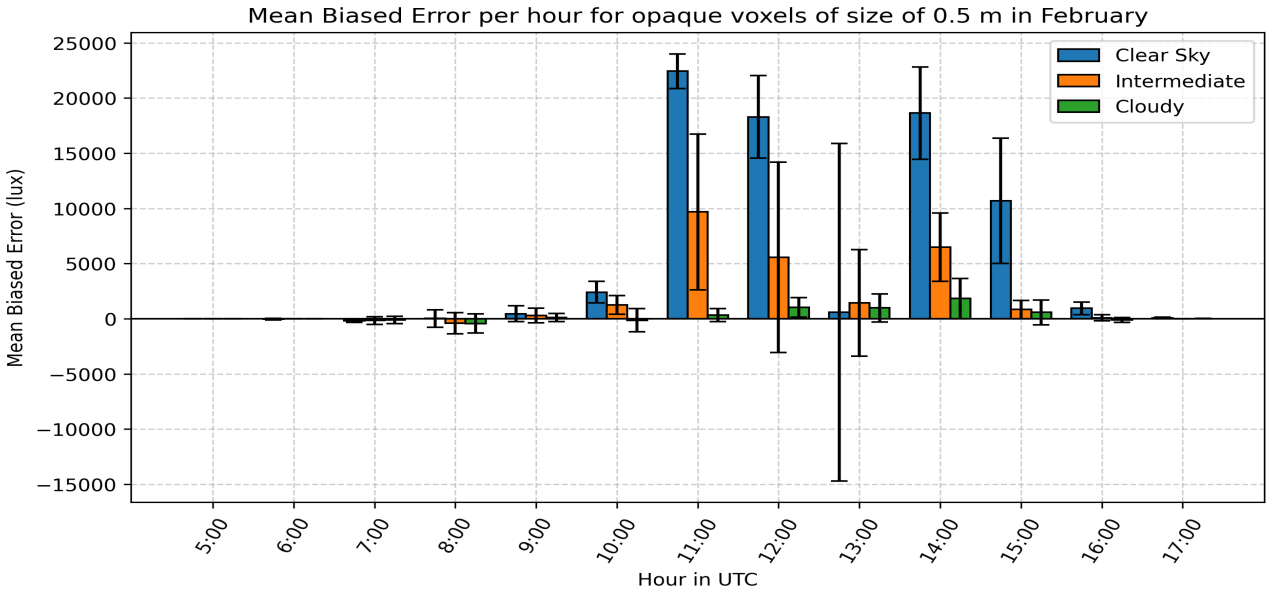


Figure 5.36.: Time series of Mean Biased Error with standard deviation per timestamp for opaque voxels of 0.5 m in February under clear sky, cloudy, and intermediate conditions.

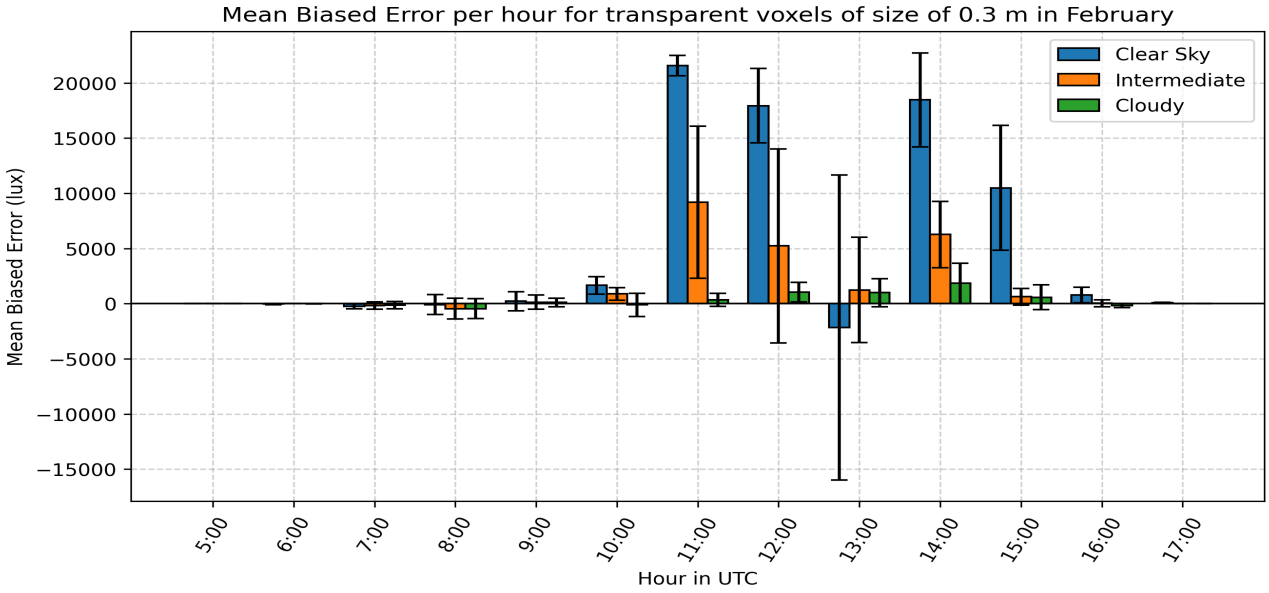


Figure 5.37.: Time series of Mean Biased Error with standard deviation per timestamp for transparent voxels of 0.3 m in February under clear sky, cloudy, and intermediate conditions.

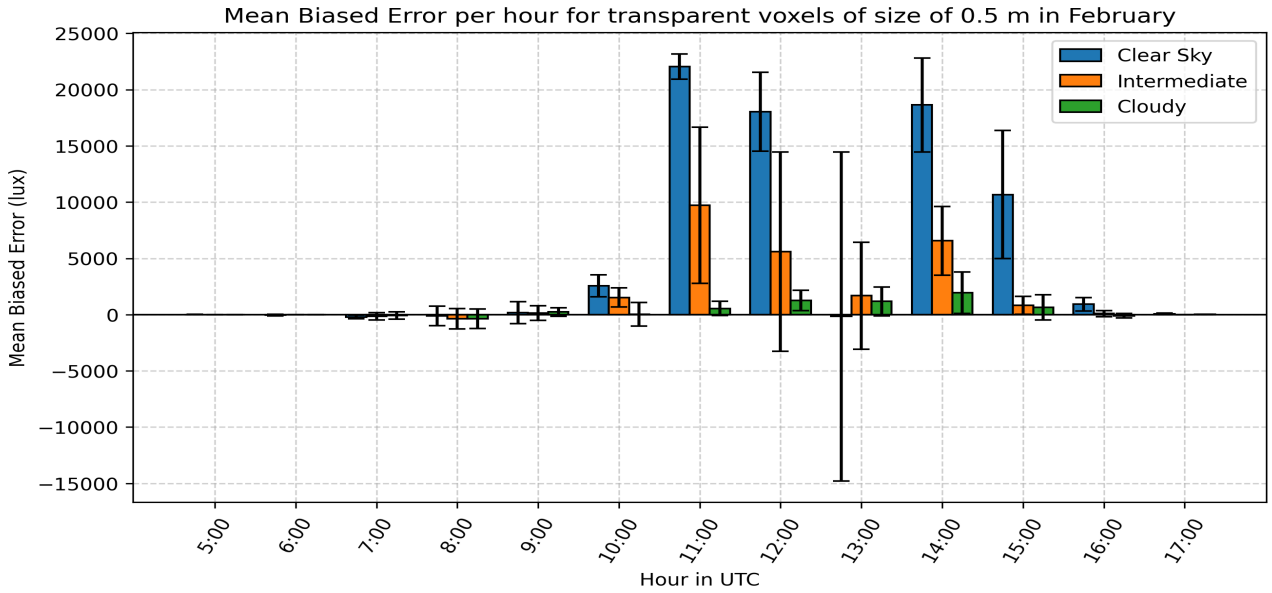


Figure 5.38.: Time series of Mean Biased Error with standard deviation per timestamp for transparent voxels of 0.5 m in February under clear sky, cloudy, and intermediate conditions.

To complement the analysis presented above, the RMSE of the MBE values for each voxel case was calculated to evaluate how close the errors approach zero and to determine whether one or multiple voxel grid configurations that performed well for the overall monthly sample also perform consistently across different sky conditions. The results are presented in Table 5.2. For clear sky and intermediate conditions, the transparent voxel grids with sizes of 0.07 m and 0.1 m yield the smallest RMSE values (9691.37 lux and 8609.63 lux for the clear sky conditions and 2955.98 lux and 2560.76 lux for the intermediate sky conditions) , which aligns with the initial findings from the full time series. In contrast, for cloudy conditions, the most suitable configurations are the opaque voxels with sizes of 0.1 m and 0.3 m (633.28 lux and 669.74 lux for the intermediate sky conditions). The voxel size of 0.1 m, or a size close to it, proves effective under all sky conditions.

Table 5.2.: RMSE of MBE values per voxel size and material under different sky conditions (in lux) in February.

			Clear sky			
Material	0.03 m	0.05 m	0.07 m	0.1 m	0.3 m	0.5 m
Transparent	16741.47	11079.33	**9691.37**	**8609.63**	9790.57	9922.81
Opaque	16261.37	10770.65	9818.54	9747.62	9954.52	10030.00
			Cloudy			
Transparent	1831.49	1225.61	921.11	909.41	692.58	766.83
Opaque	1737.41	1094.04	834.34	**633.28**	**669.74**	685.16
			Intermediate			
Transparent	6268.63	3729.79	**2955.98**	**2560.76**	3445.00	3665.13
Opaque	5716.27	3303.43	2989.39	3072.98	3588.14	3632.39

5.2. June results overview

RMSE per June Case		
Point Cloud Case		
12634.49		
Alpha Shape Case		
9793.44		
Voxel Grid Case		
Sizes (m)	1st approach	2nd approach
0.03	11690.99	12010.18
0.05	9588.92	9945.00
0.07	9555.12	10315.16
0.10	9741.06	9896.88
0.30	9869.44	9899.31
0.50	9368.68	9655.70
Convex Hull Case		
7885.72		

MBE per June Case		
Point Cloud Case		
-4337.18		
Alpha Shape Case		
1668.56		
Voxel Grid Case		
Sizes (m)	1st approach	2nd approach
0.03	-2960.44	-3396.45
0.05	-683.79	-1355.18
0.07	312.57	-1793.85
0.10	1129.29	-474.26
0.30	1570.62	1135.84
0.50	1832.69	1321.15
Convex Hull Case		
3021.98		

Figure 5.39.: RMSE and MBE values for June simulations.

Comparing the statistics derived from the June simulations in **Figure 5.39** with those from the February simulations in **Figure 5.1**, it is evident that the February simulations performed better overall, as indicated by their lower RMSE and MBE values. This suggests that the geometric representations created for June simulations from this research, together with the materials assigned to them, do not model trees as accurately as as those used in the February simulations.
A closer look at **Figure 5.39** demonstrates that the representation from the Point Cloud Case led to the strongest overestimation of illuminance values with a MBE of -4337.18 lux, similar to the corresponding February simulation. This shows neither the synthesized point cloud has adequate point density to capture the trees of the study area accurately.
Also, even if it was expected that the Alpha Shape Case would be more successful in these simulations, since the produced mesh representation resemble more the real appearance of the trees with dense vegetation, it resulted in a considerable underestimation of illuminance values. This may be attributed to the unsuitable opaque material, that was assigned to the mesh representations. It is likely that these mesh representations acted as a significant obstacle to incident light.
About the Convex Hull Case, the RMSE and MBE values can give some interesting insights about its simulated illuminance values. At first, it shows the largest positive bias (3021.98 lux), which highlights that the 3D object constructed from the convex hull of the branch points essentially blocks incident light, due to its broad form and the expansion of the leaf material across its surface. Interestingly, however, this case has the smallest RMSE value (7885.72 lux) in **Figure 5.39**. The relatively low deviation of errors between the sensor records and the simulation values accounts for this RMSE, which clearly does not indicate good simulation performance when MBE is taken into account.
In regards to the Voxel Grid Case, although the RMSE and MBE values are larger compared to those of the February simulations, the June simulations exhibit similar trends in their results. At first, in both cases voxels of 0.03 m and 0.05 contribute to highly overestimated illuminance values, which apparently improve to a small extent in the first approach and at 0.05 m achieving MBE value of -683.79 lux. It is apparent from the MBE values that the highest level of overestimation occurs in the second approach. On the other hand, again

in both fist and second approach, at the sizes of 0.30 and 0.50 m the simulated illuminance values were much less than expected because of the high RMSE an MBE values, which exceed the corresponding values of other sizes. Also, it is proven that in the first approach simulated values were underestimated to a larger extent.
Next, focusing on the intermediate size, in the first approach the size of 0.07 led to a RMSE of 9555.12 and a MBE of 312.57 lux, explaining that the simulated illuminance values are better aligned with the sensor data, following a relatively small upward trend while still exhibiting some outlier errors due to the high RMSE. A good level of fit between the simulated values and the sensor data is observed in the second approach with a size of 0.10 m, although the errors follow a downward trend, as indicated by an RMSE of 9896.88 lux and an MBE of -474.26 lux. In general, the high RMSE values across all cases (greater than 9000 lux) suggest that extreme error values are disproportionately influencing the RMSE calculation.

5.2.1. Whole Time Series Analysis — Voxel Grid Case

Based on the MBE values **Figure 5.39** in the first approach, although the simulated values were highly overestimated with a MBE of -2960.44 lux at the voxel size of 0.03 m, a slight rise of the voxel size by 2 cm increase rapidly the MBE value to -683.79 lux. Next, from the voxel size of 0.07 m onward, the simulated values were underestimated, with the increasing trend to decline, as the voxel size rises. In terms of deviation between the simulated values and the sensor data, the better performance is observed at the size of 0.07 m with a MBE of 312.57 lux, but with some large errors as indicated from the high RMSE value.
This behavior can be explained with the scarcity and the voxel size in each case. From **Figures 5.40** to **5.45** the nearest voxel distances distributions are demonstrated. At first, the distribution of nearest voxel distances for the June 2023 case with a voxel size of 0.03 m (**Figure 5.40**) is strongly right-skewed. Even if the most distances concentrated between 0.000 m and 0.114 m, after this range there is a considerable number of large distances pull the mean upward to 0.099 m. The difference from **Figure 5.2** can be attributed to the fact that the point cloud has become denser only in certain areas, particularly in the upper parts of the trees, rather than across the entire study area, while the points representing the stems remain intact. Nevertheless, the high concentration of voxels at small distances could not compensate for the small voxel size, resulting in insufficient blocking of light by these voxels and, consequently, in high simulated illuminance values.
Moving to the nearest voxel distance distribution at the size of 0.05 m (**Figure 5.41**), a similar pattern to the previous case is observed, but with fewer large distances. However, although most voxel distances remain below 0.023 m, reflecting again the points on the tree branches, the mean distance shifts to 0.082 m. The significant improvement in the MBE value can be attributed to the 2 cm increase in voxel size, which produces a slightly more meaningful obstacle to the incident light.
Regarding the case of 0.07 m (**Figure 5.42**) , its nearest voxel distance distribution remains right-skewed but is more compressed toward smaller distances. Most distances fall within the range of 0.000–0.043 m, while the mean distance is 0.067 m, which is closer to the predominant distance range compared to the previous cases. This distribution of voxels in space, together with the given voxel size, suggests a more solid 3D object blocking the incident light. Moreover, due to the favorable MBE value, this balance between voxel spacing and voxel size (capturing both porosity and solidity) leads to good performance in this case.
In the case of 0.1 m, where the excessive underestimation of simulated illuminance values begins, a different behavior in the distribution is observed in **Figure 5.43**. Many more voxels have their nearest neighbor within 0.02 m, with only a few exceptions that are significantly less numerous than in the previous cases. Finally, for the voxel sizes of 0.30 m and 0.50 m (**Figures 5.44** and **5.45**), almost all voxels are within 0.013 m of each other. Combined with the large voxel sizes, this results in a broad and coarse tree representation with little preserved detail, which prevents a suitable fraction of light from reaching the simulation target.

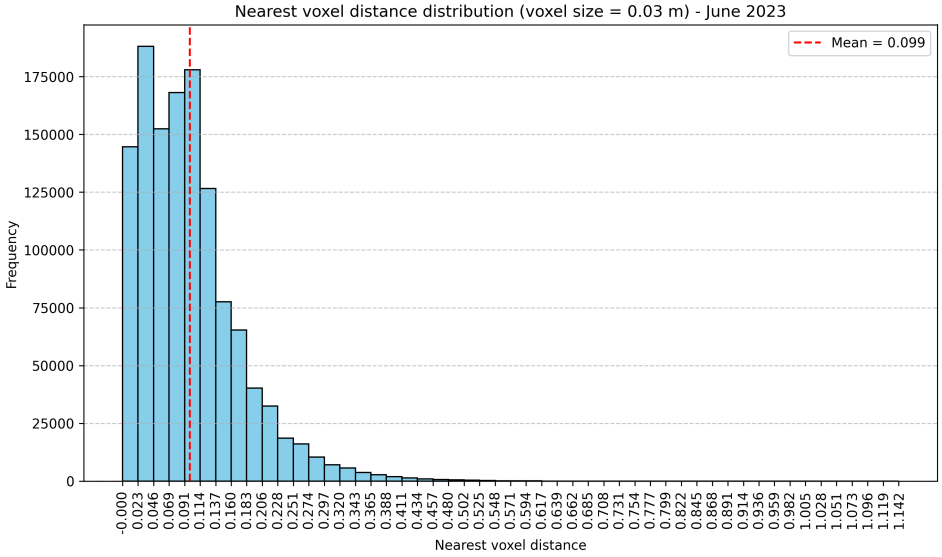


Figure 5.40.: Nearest voxel distance distribution at size 0.03 m in June.

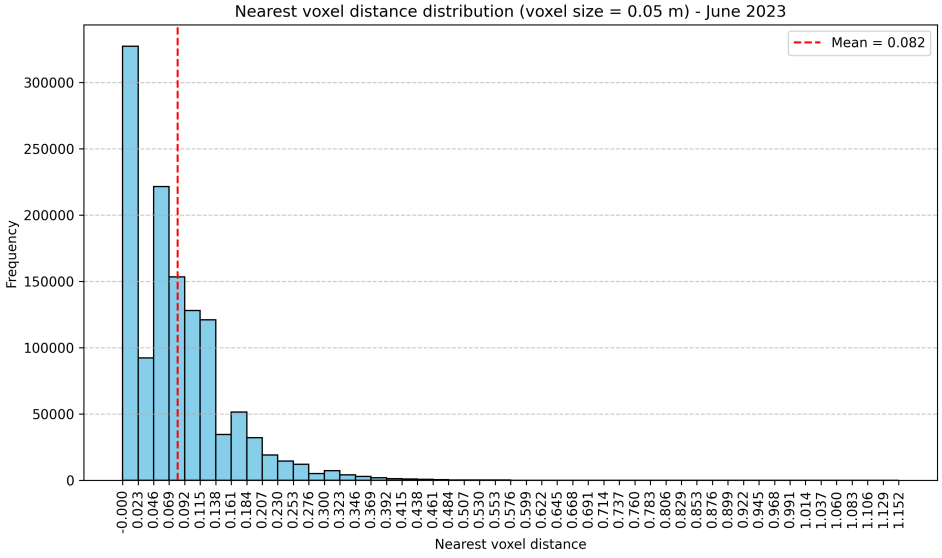


Figure 5.41.: Nearest voxel distance distribution at size 0.05 m in June.

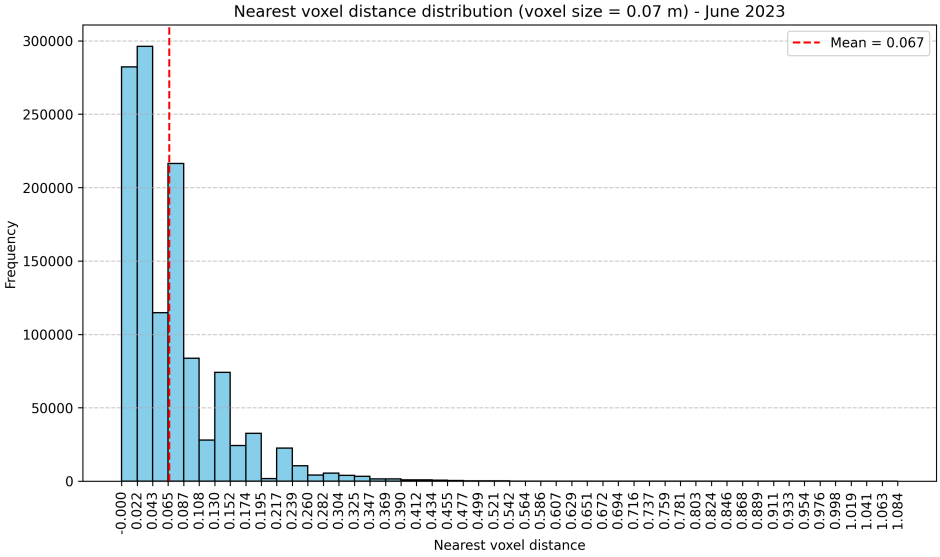


Figure 5.42.: Nearest voxel distance distribution at size 0.07 m in June.

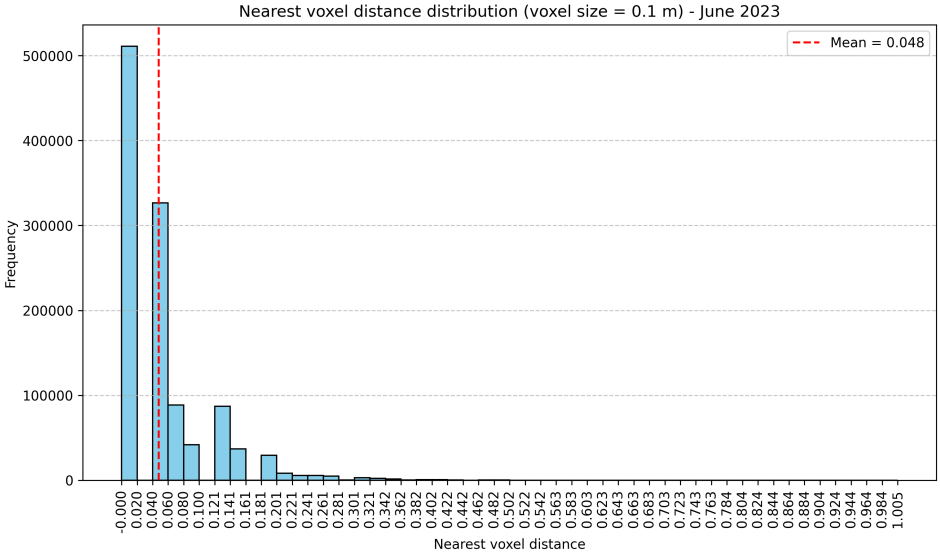


Figure 5.43.: Nearest voxel distance distribution at size 0.10 m in June.

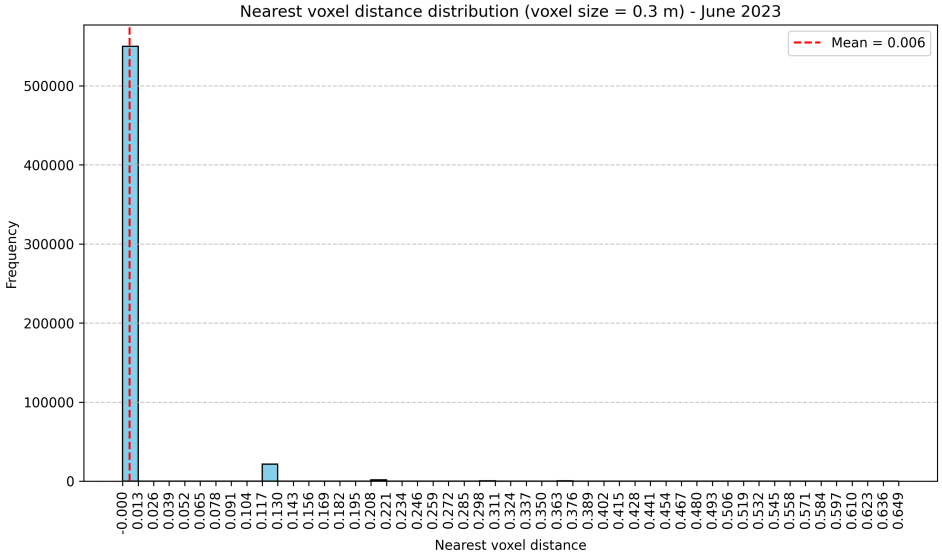


Figure 5.44.: Nearest voxel distance distribution at size 0.30 m in June.

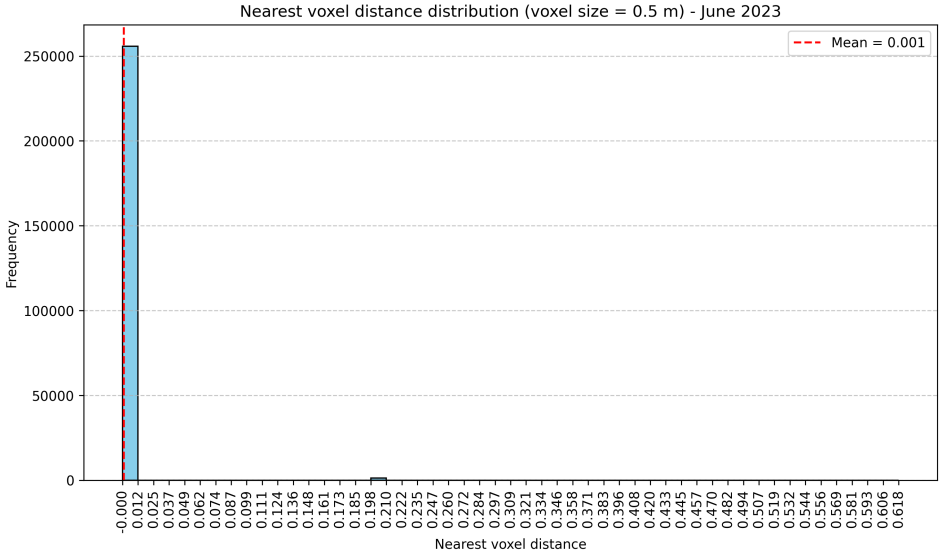


Figure 5.45.: Nearest voxel distance distribution at size 0.50 m in June.

Next, the error distribution histograms for each voxel size are shown in **Figures 5.46** to **5.51**. At the voxel size of 0.03 m, errors are spread across a wide range around the mean. The distribution leans more toward larger negative values, consistent with the mean value of -2960.44 lux and the strong overestimation in the simulation results. Then, regarding the corresponding distribution of 0.05 m, even though MBE and standard

deviation has been improved, there is still significant overestimation. Subsequently, the distribution of 0.07 m is more centered and balanced compared to the previous sizes given the relatively small mean error (312.57 lux) with a slight tendency toward the larger positive errors. Finally, as the voxel size increases, the mean value shifts and the distribution becomes increasingly right-skewed, reflecting the growing underestimation of the simulated values.

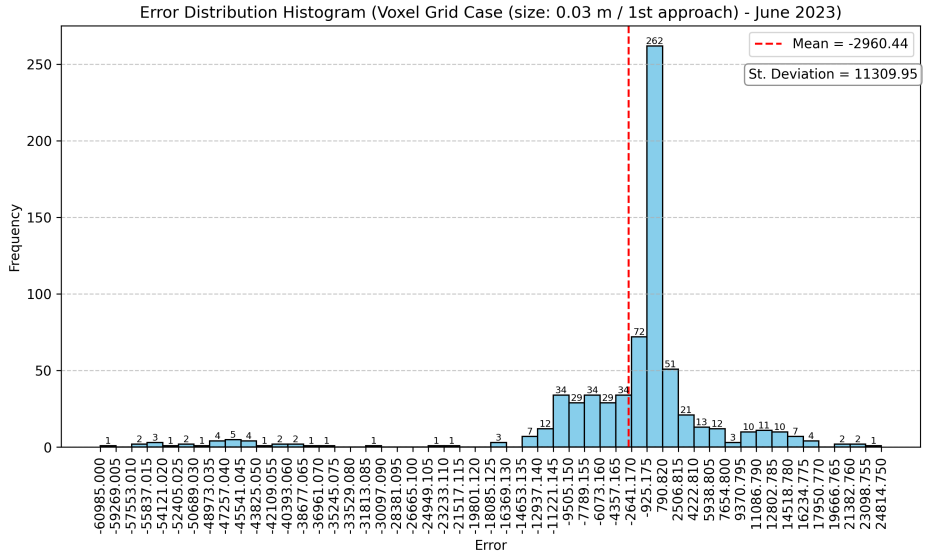


Figure 5.46.: Error distribution histogram at size 0.03 m based on the first approach in June.

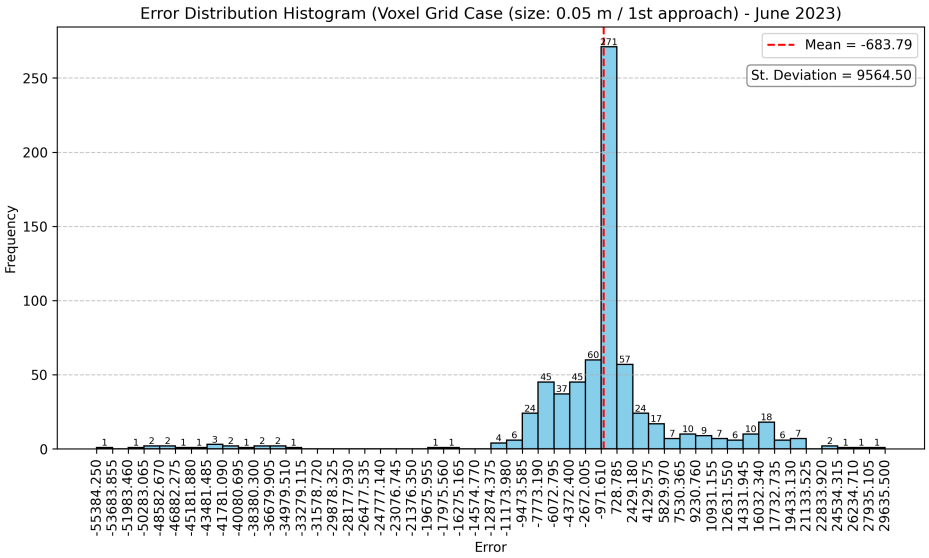


Figure 5.47.: Error distribution histogram at size 0.05 m based on the first approach in June.

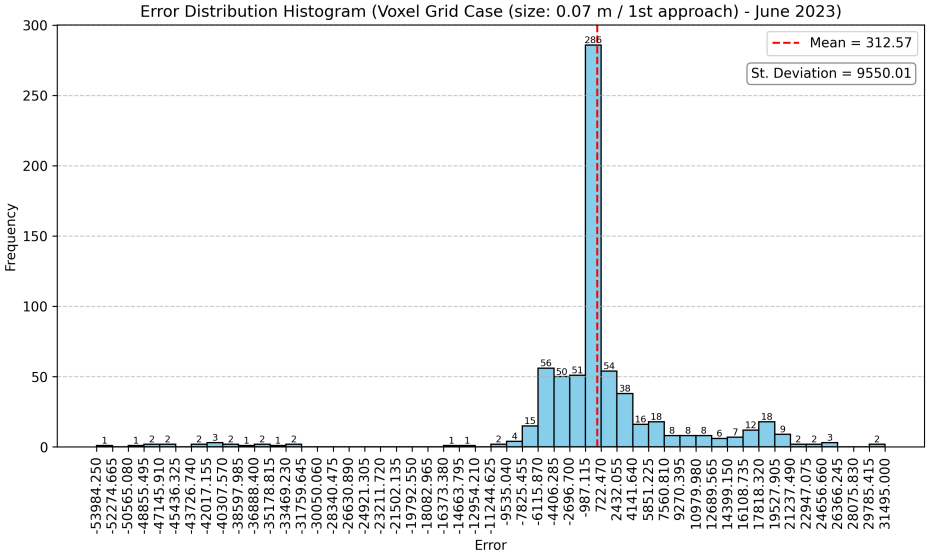


Figure 5.48.: Error distribution histogram at size 0.07 m based on the first approach in June.

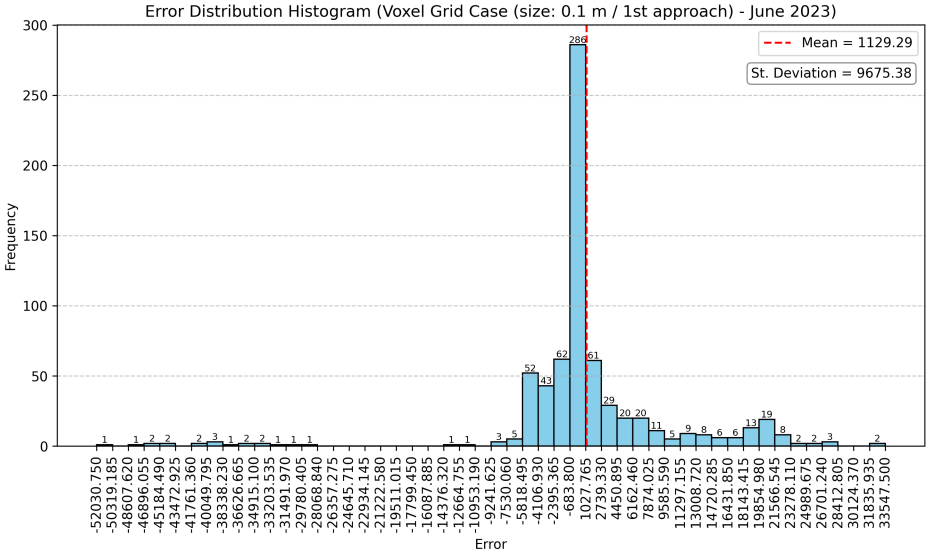


Figure 5.49.: Error distribution histogram at size 0.10 m based on the first approach in June.

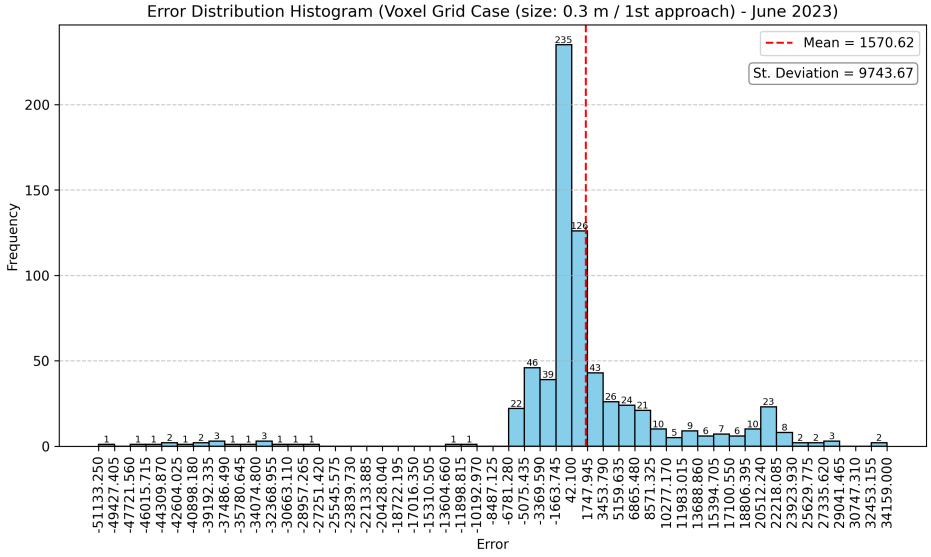


Figure 5.50.: Error distribution histogram at size 0.30 m based on the first approach in June.

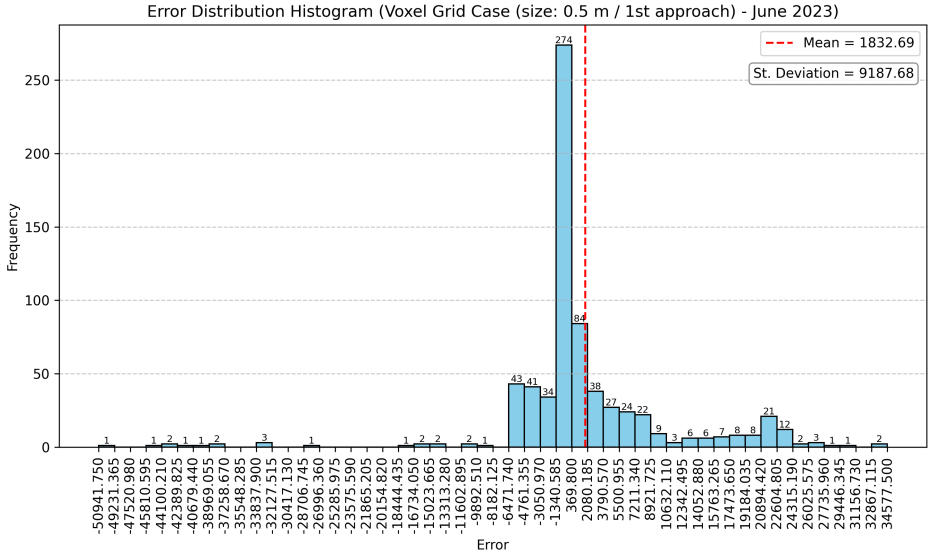


Figure 5.51.: Error distribution histogram at size 0.50 m based on the first approach in June.

Regarding the second approach, at the voxel size of 0.03 m the MBE value of -3396.45 lux indicates a high level of overestimation of the simulated illuminance values. This tendency persists at 0.05 m, though to a lesser extent, with an MBE of -1355.18 lux. At 0.07 m, however, the MBE unexpectedly increases again to -1793.85. The simultaneously higher RMSE compared to the smaller voxel sizes indicates a considerable number of large errors between the simulation results and the sensor data, rendering this case less reliable than the first approach. The best performance in this approach is observed at 0.10 m, with an MBE of -474.26 lux, suggesting a relatively small level of overestimation. Finally, at voxel sizes from 0.30 m to 0.50 m, the

simulated illuminance values are underestimated, with the degree of underestimation increasing as voxel size rises.
These results can be explained not only by the point density and the voxel size in each case, as in the first approach, but also by the distribution of different voxel types in terms of their transmittance. At first, as mentioned earlier, there is a high level of overestimation at the smaller voxel sizes due to point scarcity and voxel resolution. However, **Table 5.3** provides further insight for comparing the cases, since the improved MBE at 0.05 m compared to 0.03 m can also be explained by the considerably higher number of voxels of transmittance of 38% and 12.5%. Their presence appears to moderate the overestimation of illuminance values. About the voxel size of 0.07 m, there is an unanticipated rise in uncertainty and overestimation, as indicated by the larger RMSE and the more negative MBE value compared to the 0.05 m case. This can be explained by the sharp increase in the number of voxels with a transmittance of 88% compared to the two smaller voxel sizes, while the 0.07 m size is still insufficient to contain enough light for the simulation target. At the voxel size of 0.10 m, where the simulation results are most successful, the majority of voxels have a transmittance of 88%, while a significant number have a transmittance of 63%, which helps to mitigate the level of overestimation. Finally, at the larger voxel sizes of 0.30 m and 0.50 m, the proportions of voxels with transmittance values of 88% and 63% decrease substantially, while those with transmittance values of 38% and 12.5% gradually increase. In combination with the large voxel sizes, this shift leads to a strong underestimation of illuminance values.

Table 5.3.: Voxel grid distributions by size and transparency class for June simulations.

Grids	0.03 m	0.05 m	0.07 m	0.10 m	0.30 m	0.50 m
1st (transmittance: 88 %)	998	4,255	1,171,244	1,069,268	541,576	232,092
2nd (transmittance: 63 %)	1,238,001	1,211,073	37,141	86,380	30,360	22,609
3rd (transmittance: 38 %)	5,148	16,741	1,445	711	1,763	2,178
4th (transmittance: 12.5 %)	25	282	56	6	62	105

Next, at **Figures** from **5.52** to **5.57** the error distribution histograms from the voxel sizes can be seen. It is apparent across all error distributions that several extremely large negative errors are present, which distort their overall balance.
Firstly, in regard to the voxel size of 0.03 m there is a clear tendency towards the large negative errors. Next, at 0.05 m the error distribution becomes more centered, as the mean error is closer to zero, though it still indicates a notable overall overestimation (MBE: -1355.18 lux). At 0.07 m, however, the considerable increase in large negative errors causes the distribution to skew leftward, resulting in a worse MBE of -1793.85 lux. Subsequently, at 0.1 m the distribution have the lowest mean value indicating a lower bias compared to all other sizes. However, it should be noted that many extreme outliers emerge on the right side apart from those on the left side, and therefore this cause the increase of the standard deviation. Finally, for the voxel sizes of 0.30 m and 0.50 m, the mean error shifts rightward, primarily due to the appearance of positive errors, both near the center of the distribution and among the extreme values. Nevertheless, the persistent presence of large negative errors counterbalances this tendency, preventing the distributions from being fully dominated by positive errors.

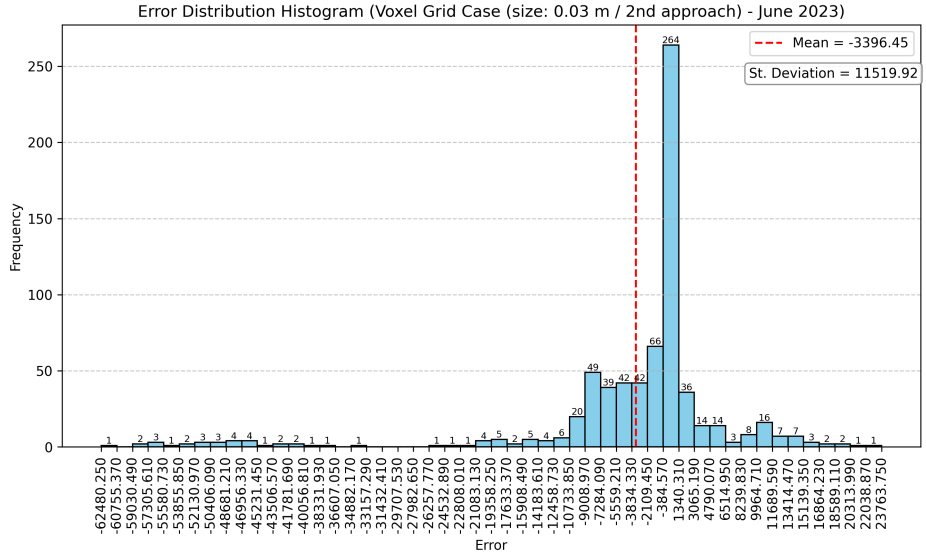


Figure 5.52.: Error distribution histogram at size 0.03 m based on the second approach in June.

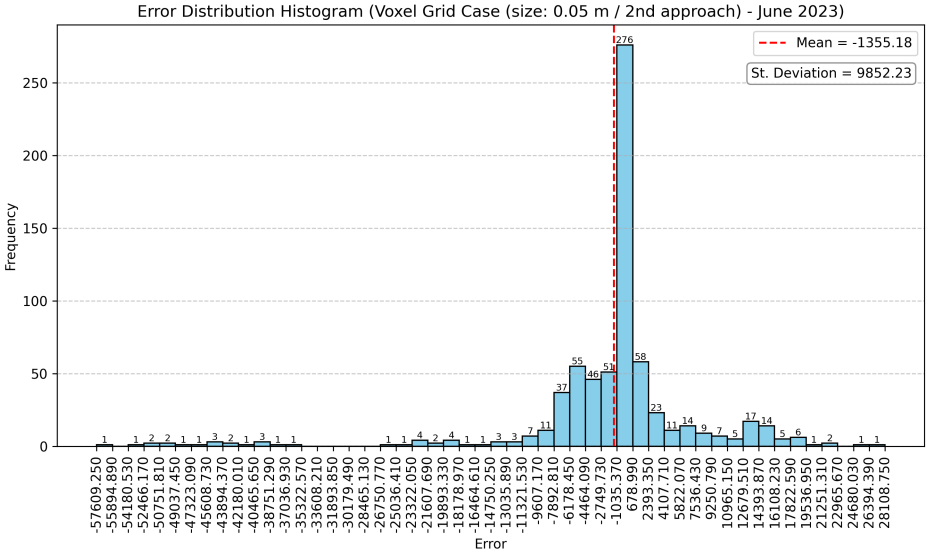


Figure 5.53.: Error distribution histogram at size 0.05 m based on the second approach in June.

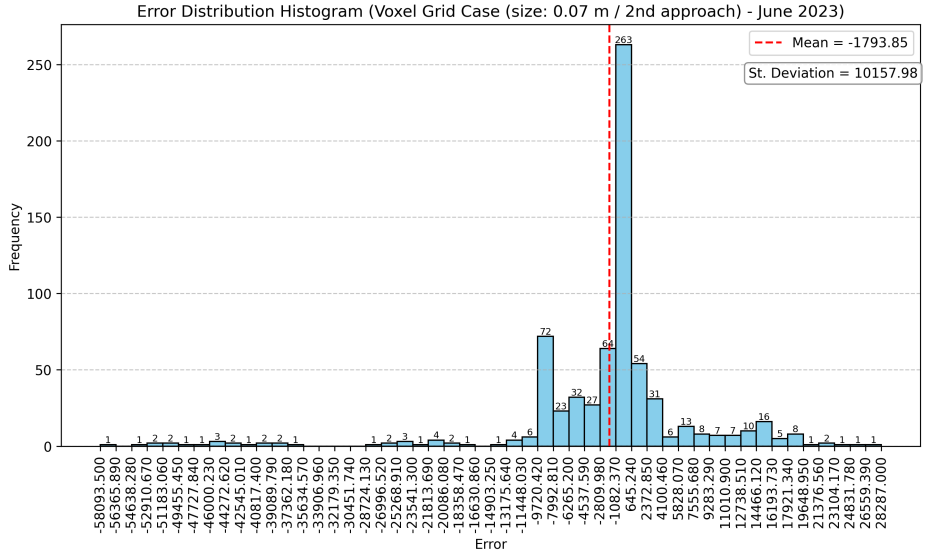


Figure 5.54.: Error distribution histogram at size 0.07 m based on the second approach in June.

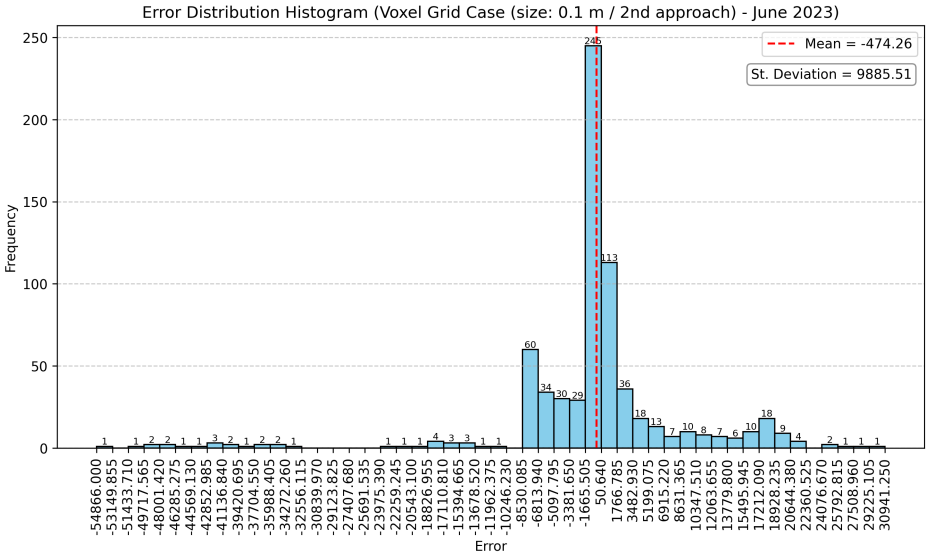


Figure 5.55.: Error distribution histogram at size 0.10 m based on the second approach in June.

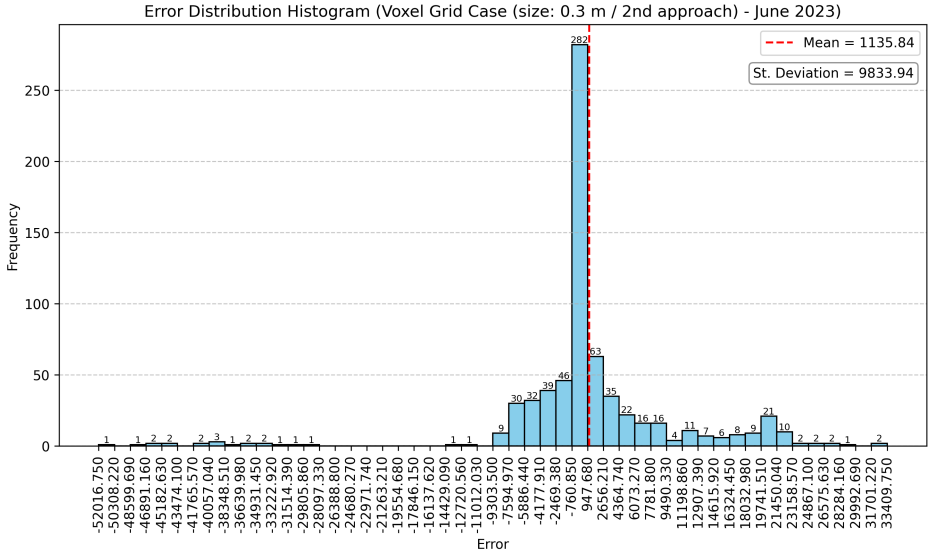


Figure 5.56.: Error distribution histogram at size 0.30 m based on the second approach in June.

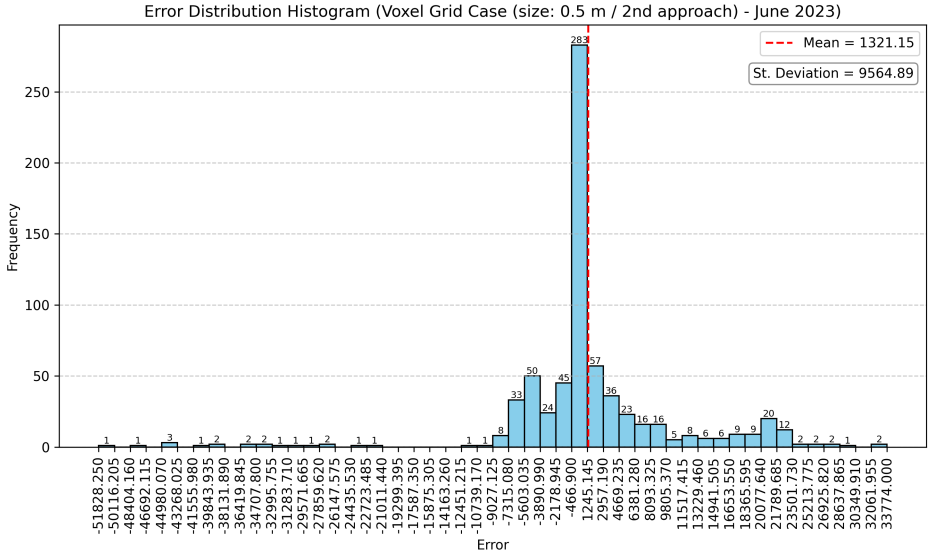


Figure 5.57.: Error distribution histogram at size 0.50 m based on the second approach in June.

5.2.2. Weather Condition Analysis — Voxel Grid Case

For June, the Voxel Case simulation results were evaluated under three different weather conditions, following the same approach used for February. The selection of the five representative days for clear sky, cloudy, and intermediate conditions was based on the same rationale applied previously. As shown in Figure 5.58, which

presents the hourly illuminance time series similarly to Figure 5.20, the five selected days for each weather condition are: (a) 3-6/6 and 14/6 for clear sky conditions, (b) 20/6, 22/6, and 27–29/6 for cloudy conditions, and (c) 1/6, 2/6, 11/6, 17/6, and 18/6 for intermediate conditions.

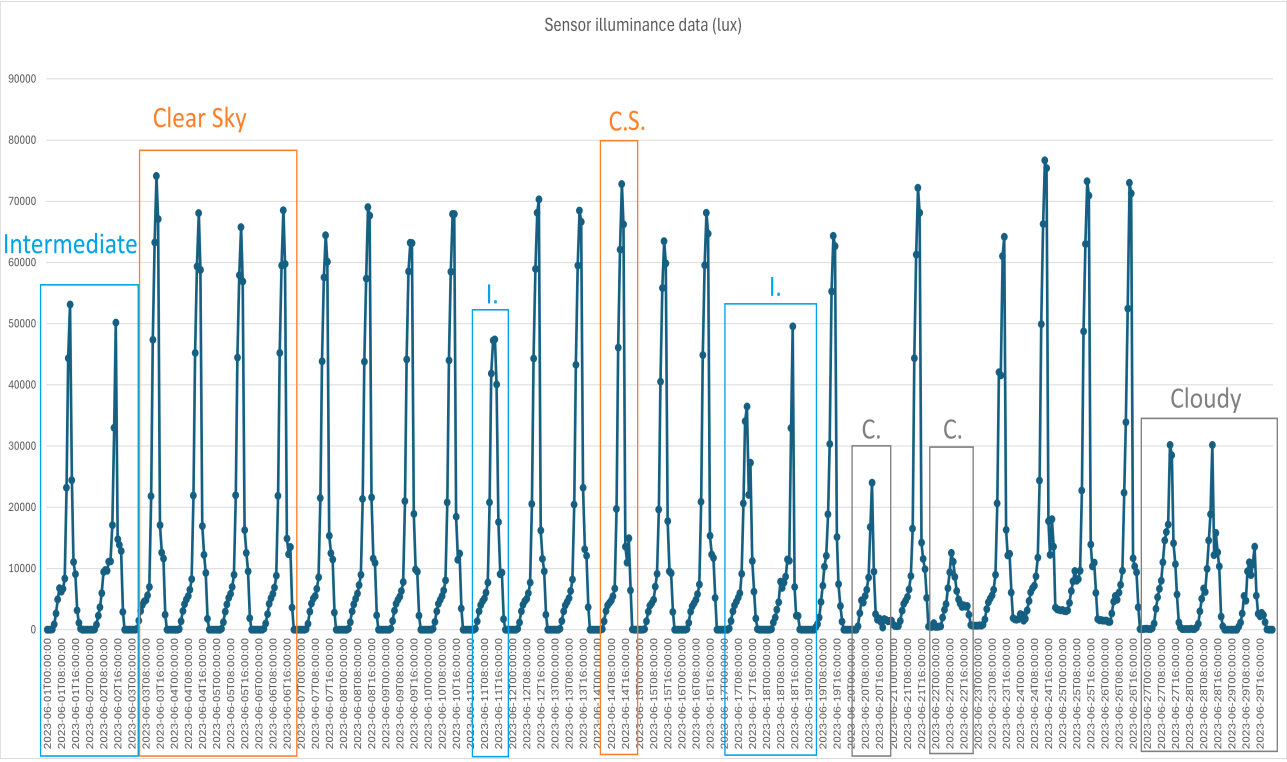


Figure 5.58.: Representative days for each sky condition in June.

According to **Figures 5.59** and **5.60**, between 06:00 and 10:00 UTC (8:00 and 12:00 local time) there is a consistent overestimation of illuminance values. This likely results from a mismatch between the modeled and actual diffuse light distribution, as the west-facing sensor is not yet directly illuminated during these hours. This overestimation is more moderate with opaque voxels, which either absorb or reflect incoming radiation.
Then, at 11:00 UTC (13:00 local time), the sensor illuminance values begin to rise, while the simulated values show a similar increase about one hour later. After this point, both sensor and simulated values form a narrow curve pattern with a one-hour shift. The sensor values drop sharply around 16:00–17:00 UTC (18:00-19:00 local time), with only the simulations using voxel sizes of 0.3 m and 0.5 m closely matching them, while the other simulations reach similar levels after 19:00 UTC (21:00 local time). The potential reason for this is the combined uncertainty in solar position calculation and sky luminance estimation, together with the influence of voxel geometry. As seen previously in February, days with high luminance are quite prone to errors.

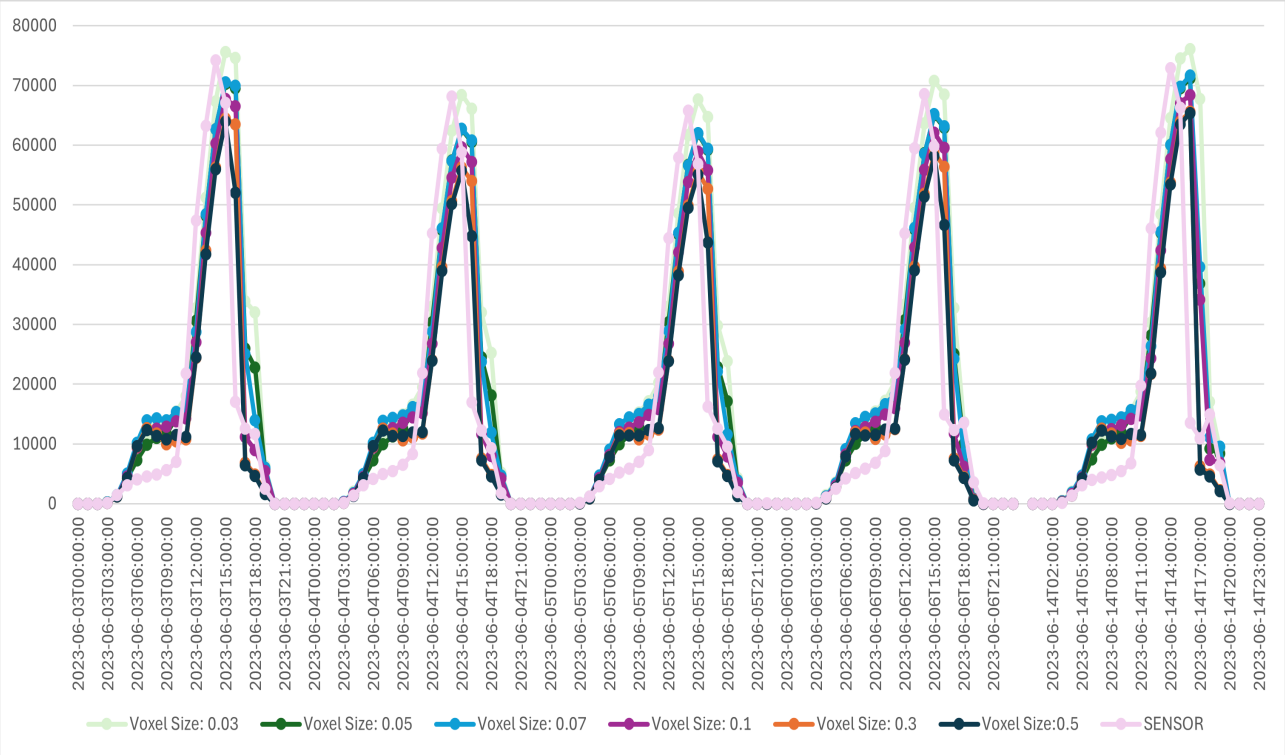


Figure 5.59.: Hourly illuminance values in lux from sensor and simulation for each voxel size using transparent voxels on five representative days under clear sky conditions in June.

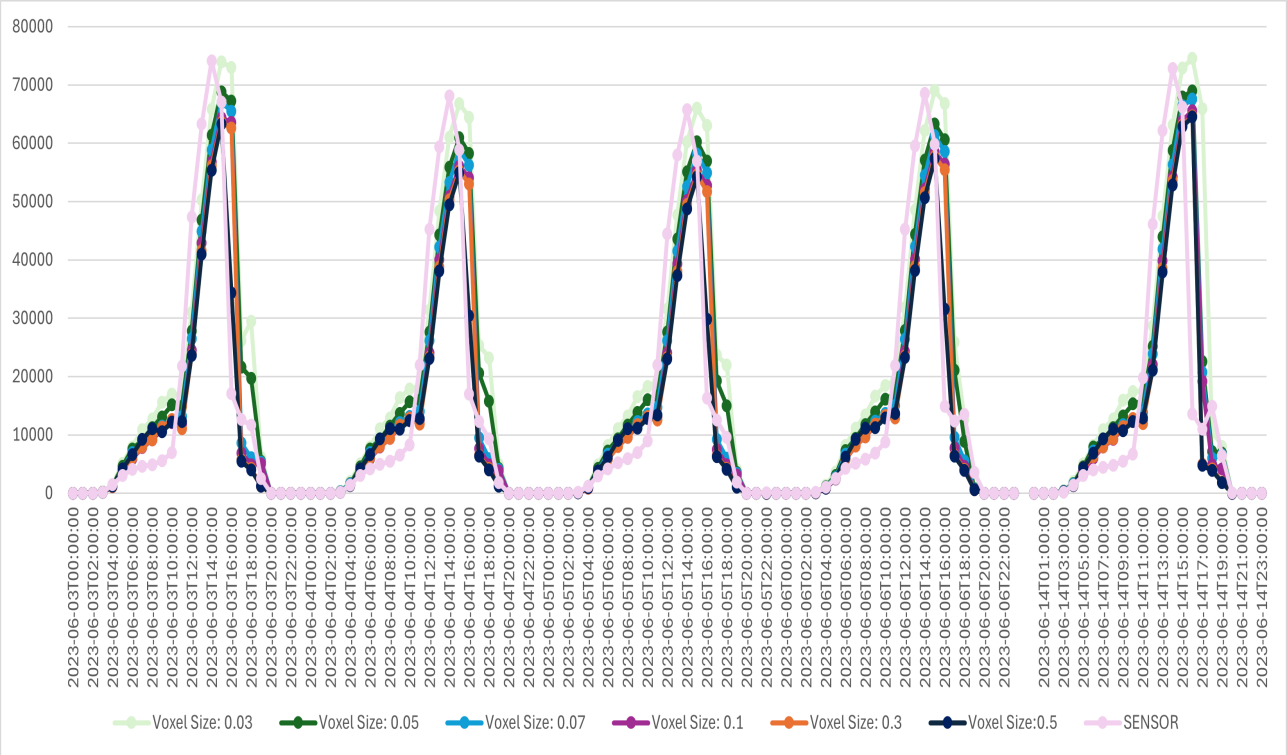


Figure 5.60.: Hourly illuminance values in lux from sensor and simulation for each voxel size using opaque voxels on five representative days under clear sky conditions in June.

Regarding the cloudy days (5.61 and 5.62), the sensor illuminance does not align strongly with any specific voxel grid model, unlike in February where voxel sizes of 0.3 m and 0.5 m performed better. This is mainly because June did not include days with consistently high cloud coverage. Nonetheless, for both materials, the simulated values corresponded more closely to the sensor data overall compared to clear sky days, except at certain times, such as on 20/6 between 7:00 and 10:00 UTC (9:00 and 12 local time), when there is overestimation and on 28/6 at 13:00 UTC (15:00 local time), when there is underestimation, where notable deviations occurred.

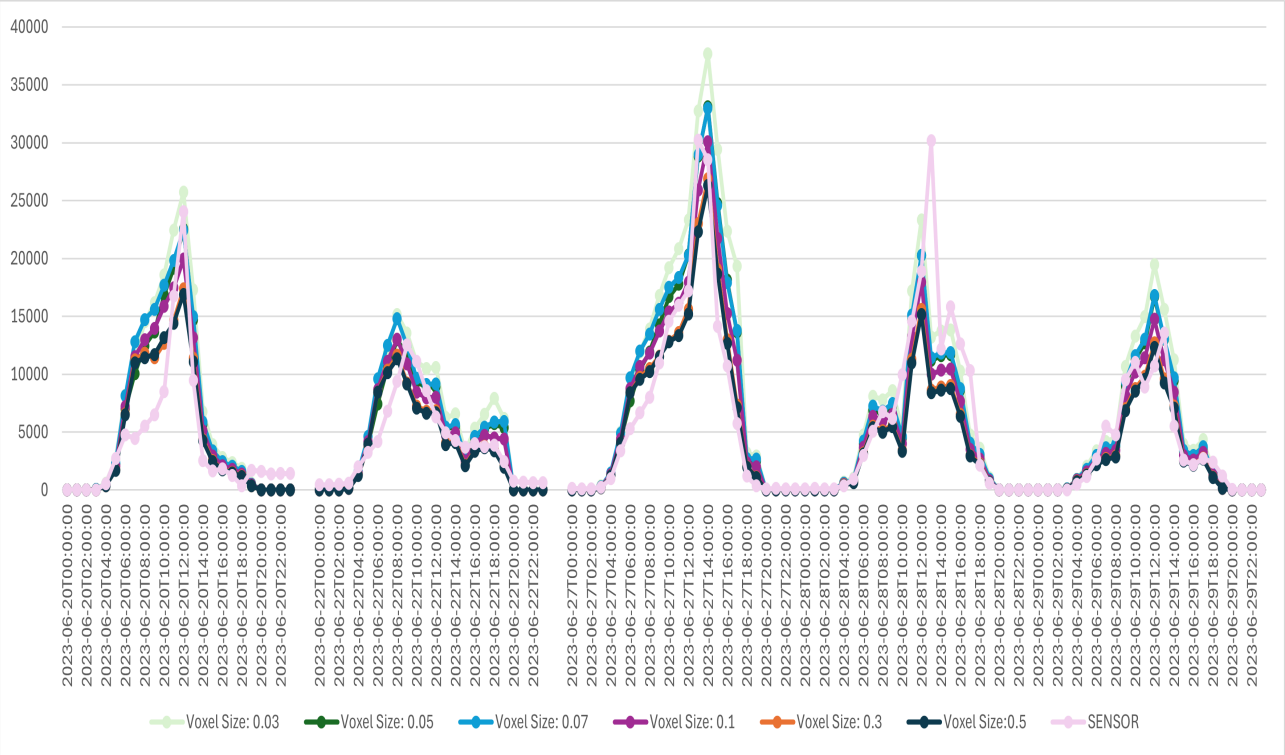


Figure 5.61.: Hourly illuminance values in lux from sensor and simulation for each voxel size using transparent voxels on five representative days under cloudy conditions in June.

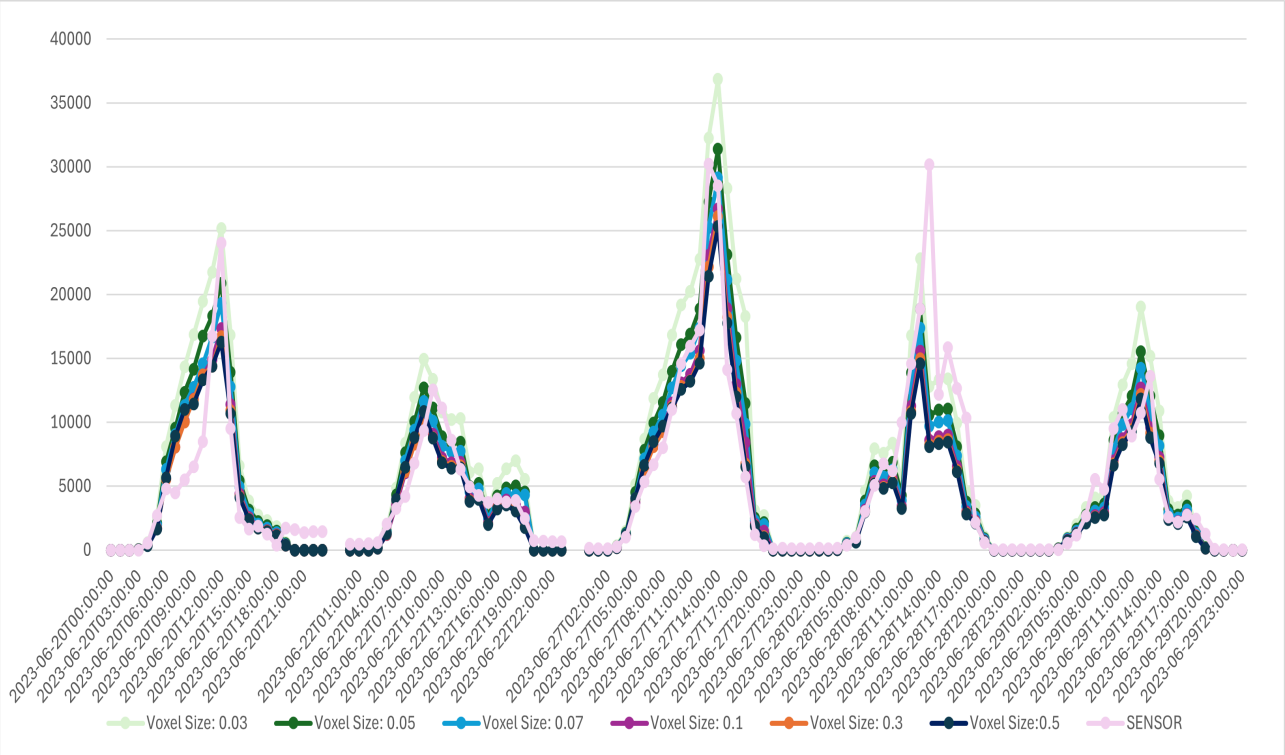


Figure 5.62.: Hourly illuminance values in lux from sensor and simulation for each voxel size using opaque voxels on five representative days under cloudy conditions in June.

Based on **Figures** 5.63 and 5.64, the consistent overestimation of illuminance values which was observed on clear sky days during the early hours, can be seen on 1/6, 11/6, and 17/6, which is less pronounced with opaque voxels. There is no specific simulation that closely aligns with the sensor data, as intermediate conditions are inherently variable and difficult to predict. However, it is worth noting that the simulations performed poorly during sudden spikes in the sensor data, such as on 2/6 at 15:00 UTC and on 18/6 at 14:00 and 15:00 UTC (16:00 and 17:00 UTC).

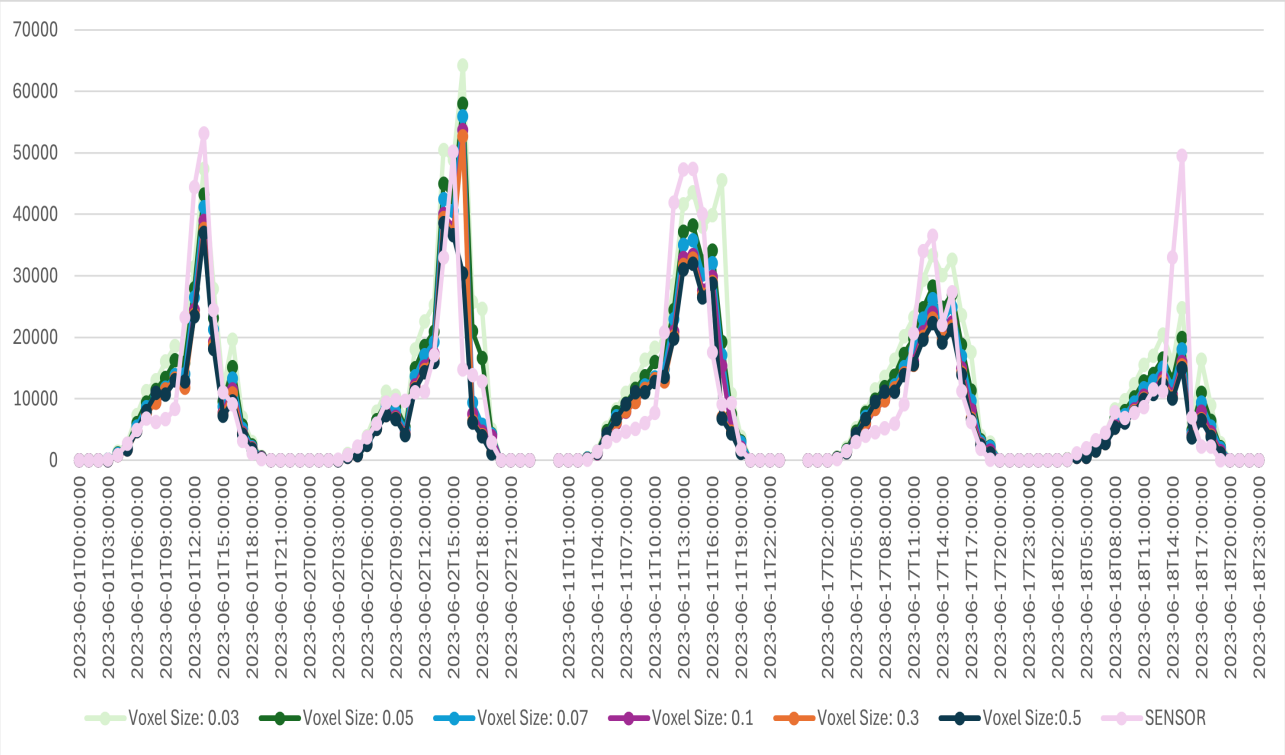


Figure 5.63.: Hourly illuminance values in lux from sensor and simulation for each voxel size using opaque voxels on five representative days under intermediate conditions in June.

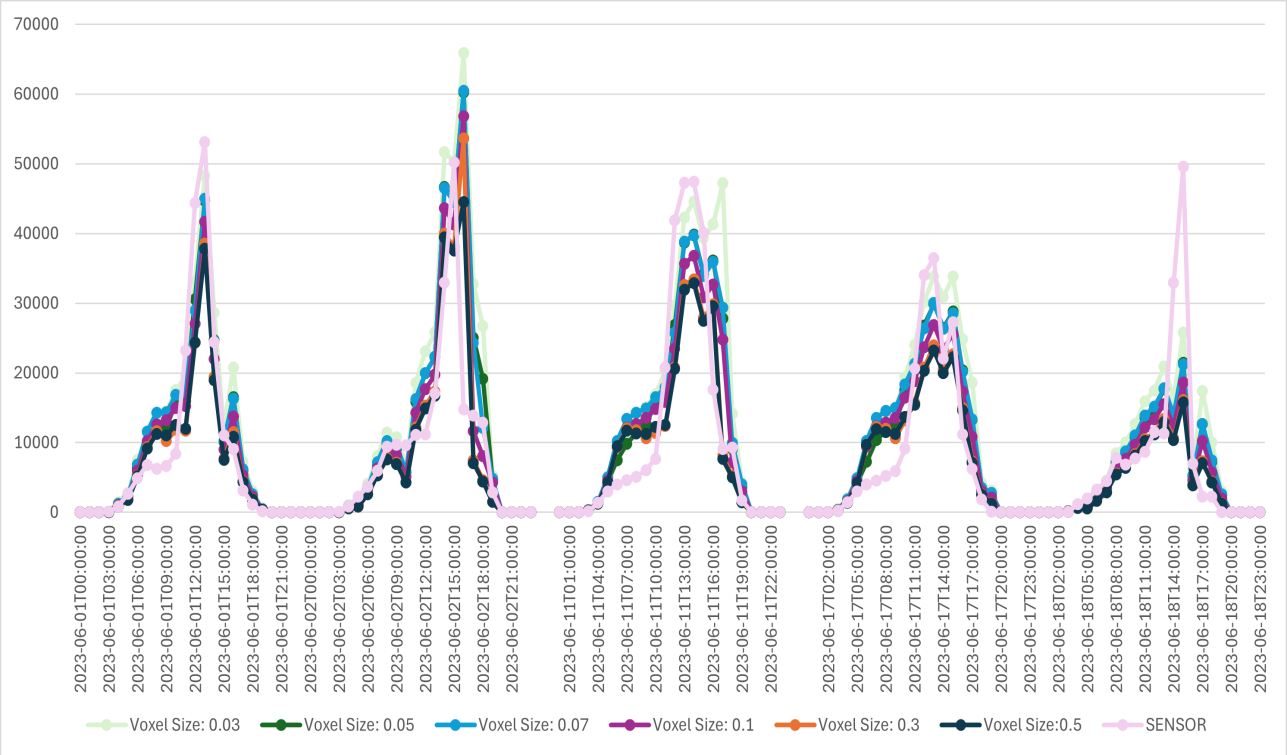


Figure 5.64.: Hourly illuminance values in lux from sensor and simulation for each voxel size using transparent voxels on five representative days under intermediate conditions in June.

Subsequently, the corresponding Mean Biased Error bar graphs for June were generated, similar to those for February, covering the hours from 2:00 to 21:00 UTC (4:00 to 23:00 local time), since the approximate daytime in the Netherlands during June is 5:30 to 22:10 local time. According to **Figures 5.65, 5.66,5.67,5.68**, it can be understood that generally errors from clear sky and intermediate conditions days follow the same pattern with the latter being smaller in magnitude, except for a handful of hours within a day. First, the overestimation of simulated illuminance values from 6:00 to 10:00 UTC (8:00 to 12:00 local time), which was observed previously, is apparent and has a clear and stable ascending trend at these diagrams. However, between 12:00 and 14:00 UTC (14:00 and 16:00 UTC), the sensor data show higher values, particularly under clear sky conditions, as solar noon occurs around 13:40 local time in June (11:40 UTC). Next, at 16:00 and 17:00 UTC (18:00 and 19:00 local time), when the fully leafed trees cast shade on the sensor, a pronounced overestimation occurs, indicating that the applied tree 3D representations are not effective during these hours. Also, under cloudy conditions, the errors are generally moderate. However, around solar noon, the simulated values become overestimated, likely due to the small voxel size and uncertainties in the modeled sky luminance. Lastly, with regards to how voxel size impacts the simulation results, it is evident that, when increasing to 0.05 m, the corresponding overestimated errors are mitigated, while the big positive errors in the midday exacerbate. Additionally, transparent voxels are prone to bring worse overestimation and less underestimation at 16:00 and 17:00 UTC (18:00 and 19:00 local time) and between 12:00 and 14:00 UTC (14:00 and 16:00 local time) respectively.

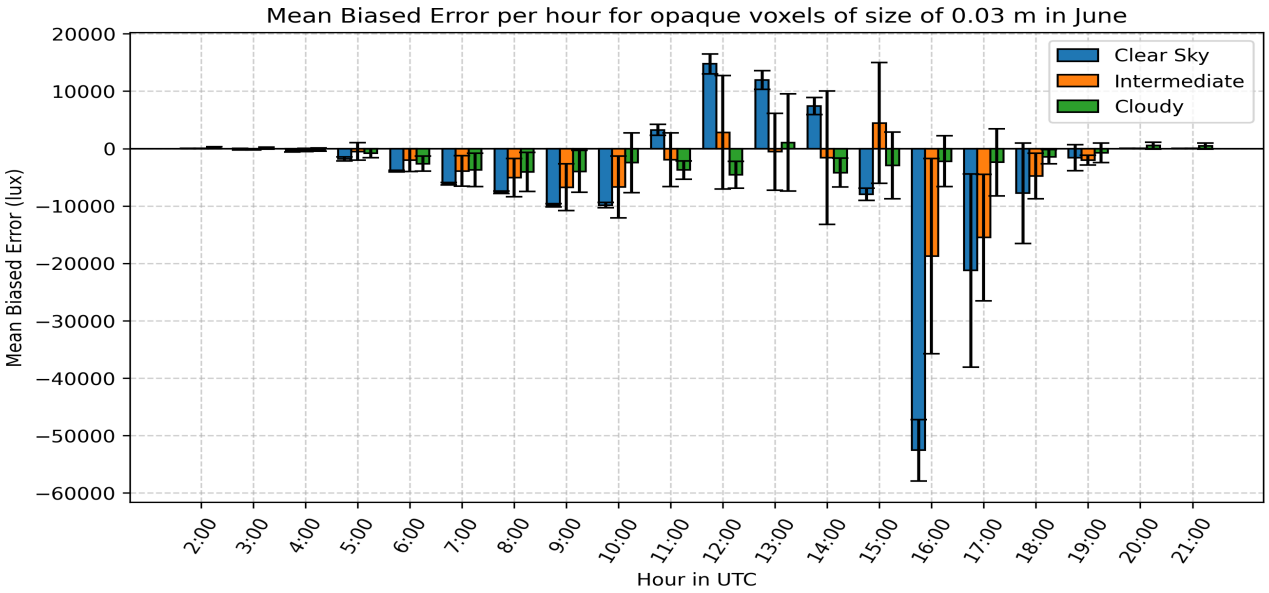


Figure 5.65.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.03 m in June under clear sky, cloudy, and intermediate conditions.

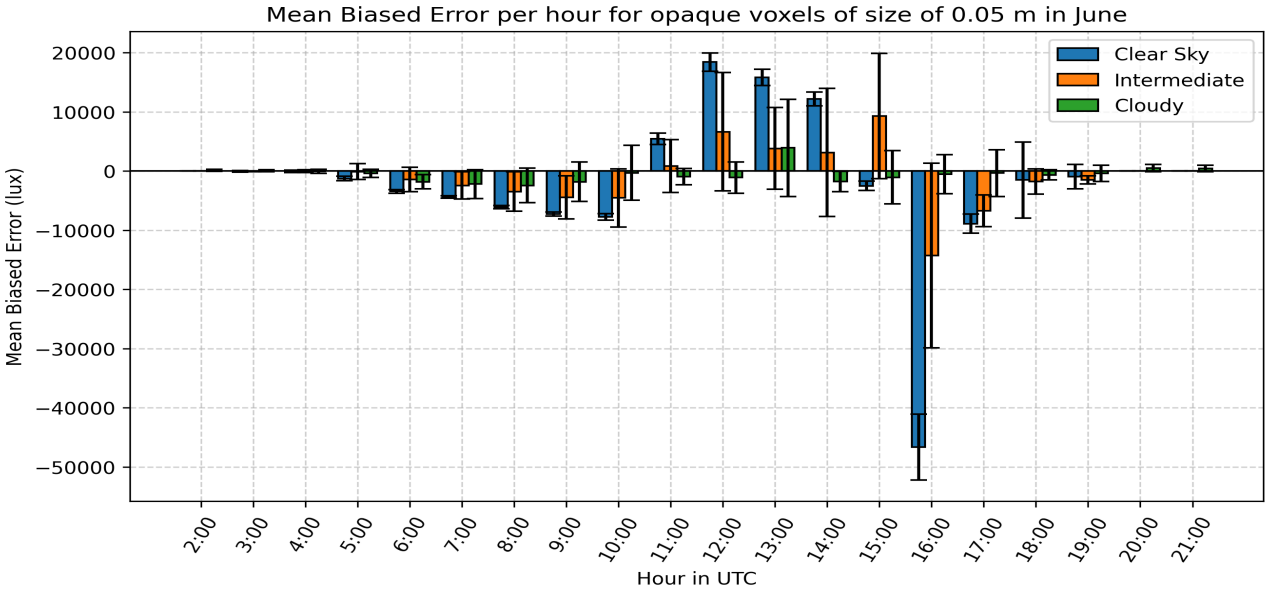


Figure 5.66.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.05 m in June under clear sky, cloudy, and intermediate conditions.

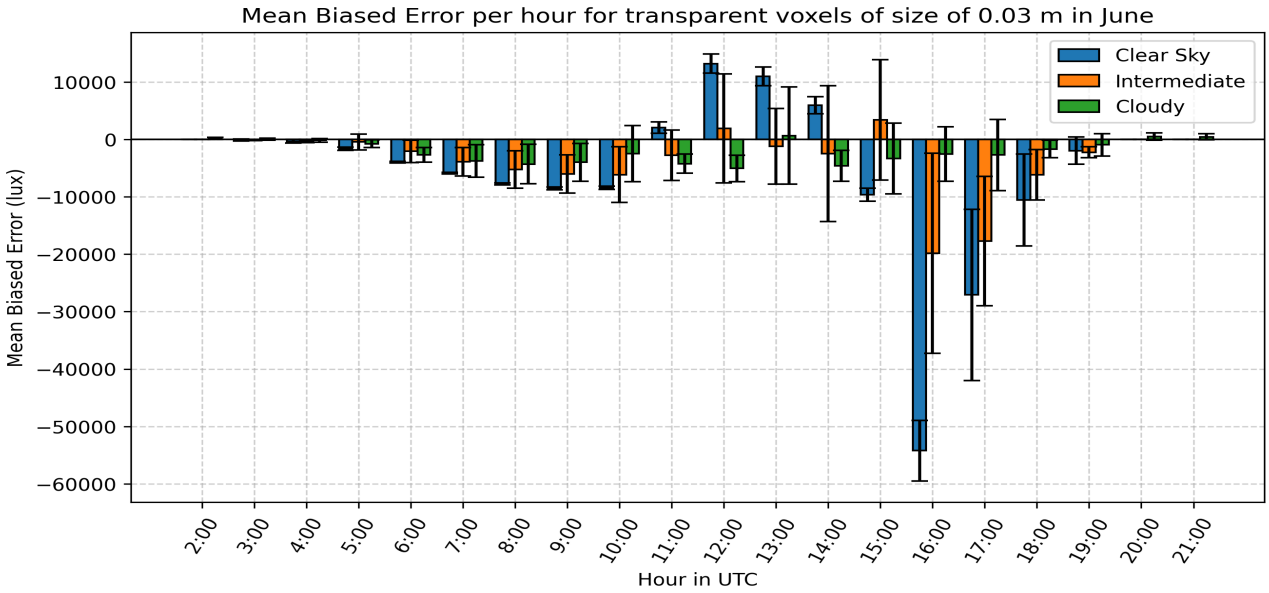


Figure 5.67.: Time series of of Mean Bias Error with standard deviation for transparent voxels of size of 0.03 m in June under clear sky, cloudy, and intermediate conditions.

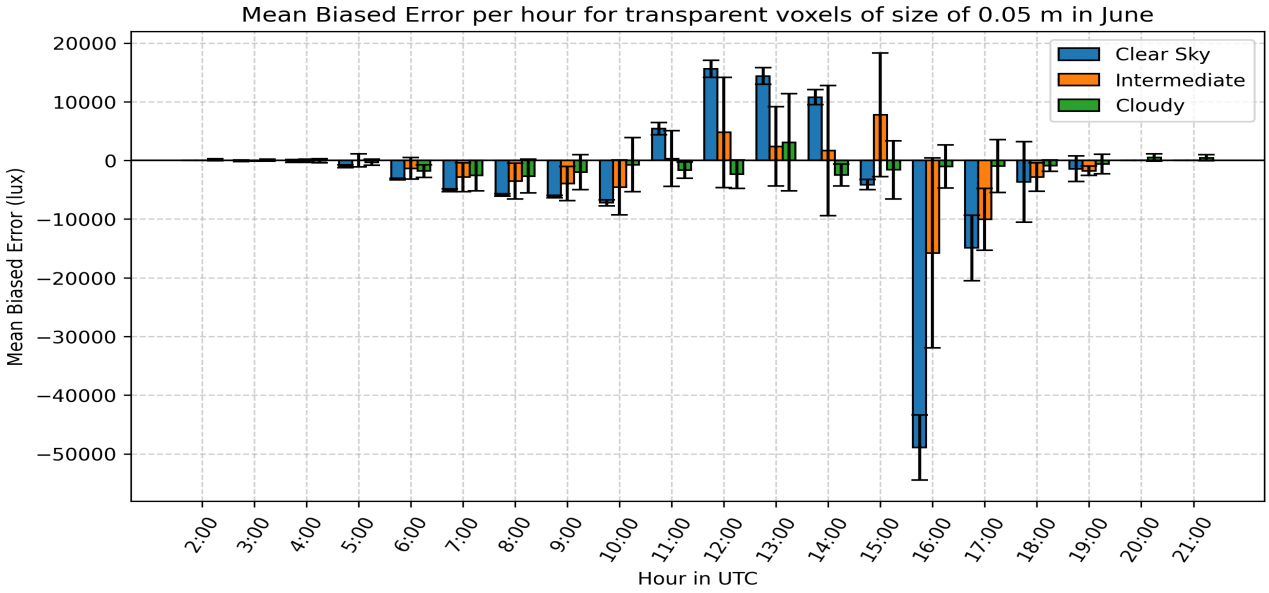


Figure 5.68.: Time series of of Mean Bias Error with standard deviation for transparent voxels of size of 0.05 m in June under clear sky, cloudy, and intermediate conditions.

Viewing **Figures 5.69, 5.70, 5.71, 5.72**, increasing the voxel size to these levels did not significantly change the previous results for clear sky and intermediate conditions. The errors are slightly reduced between 6:00–10:00 UTC (8:00-12:00 local time) and around 16:00–17:00 UTC (18:00-19:00 local time), suggesting that larger voxels are more resilient to both scattered and direct light. However, the pronounced errors between 12:00 and 14:00 UTC (14:00-16:00 local time) persist. For cloudy days, interestingly with opaque voxels, the errors around solar noon gradually increase with voxel size, while with transparent voxels, they remain relatively stable.

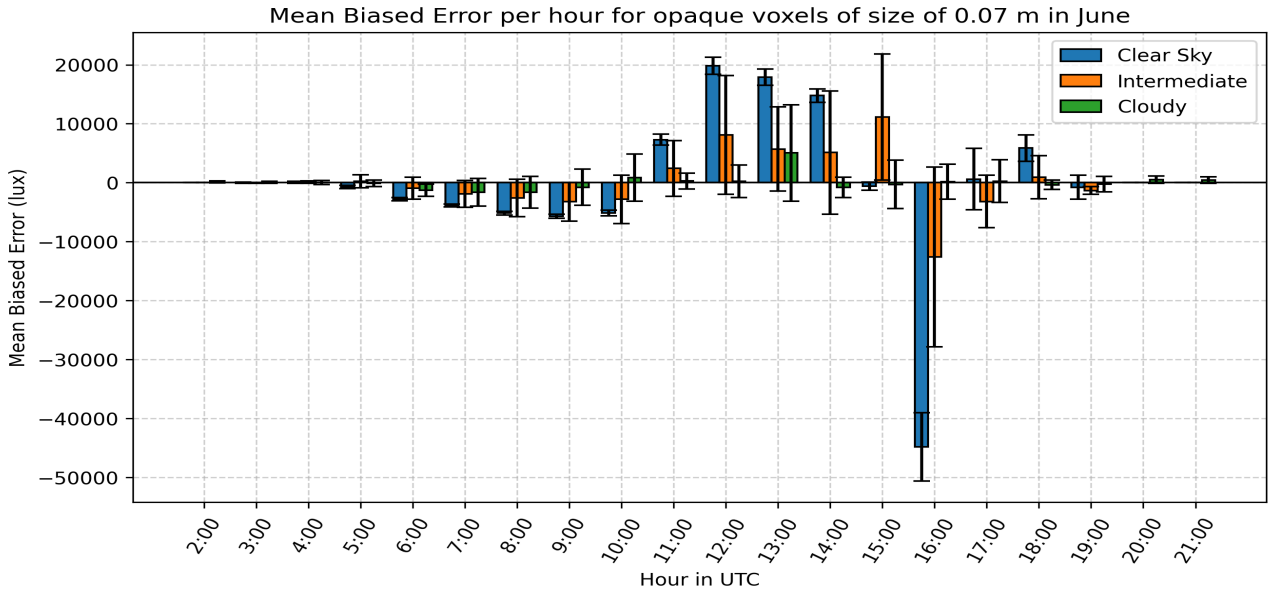


Figure 5.69.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.07 m in June under clear sky, cloudy, and intermediate conditions.

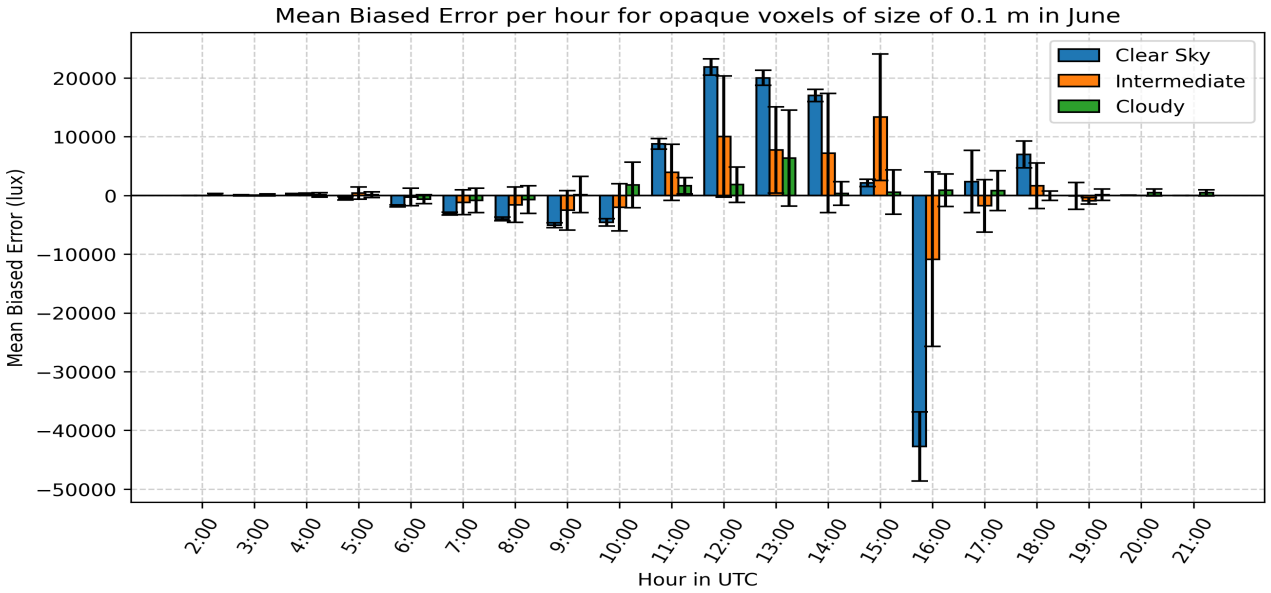


Figure 5.70.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.1 m in June under clear sky, cloudy, and intermediate conditions.

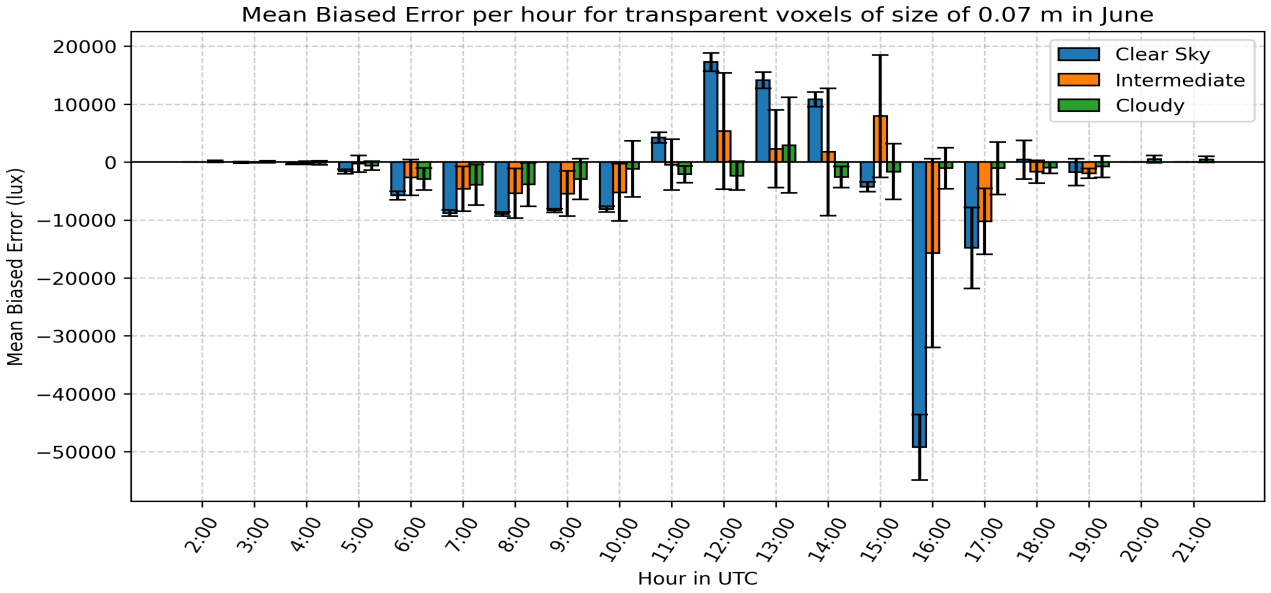


Figure 5.71.: Time series of of Mean Bias Error with standard deviation for transparent voxels of 0.07 m in June under clear sky, cloudy, and intermediate conditions.

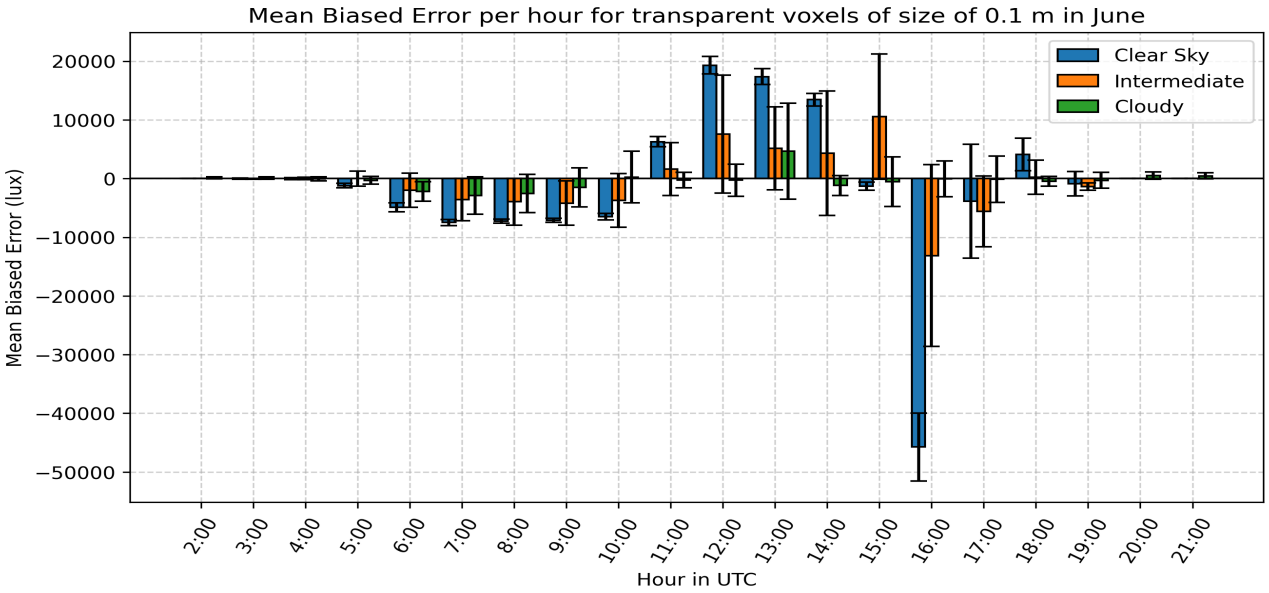


Figure 5.72.: Time series of of Mean Bias Error with standard deviation for transparent voxels of 0.1 m in June under clear sky, cloudy, and intermediate conditions.

Lastly, based on **Figures 5.73, 5.74, 5.75, 5.76**, as the voxel size increases to 0.3 and 0.5 m, the errors under clear sky and intermediate conditions between 6:00 and 10:00 UTC (8:00 and 12:00 local time) become more stable, while those between 12:00 and 13:00 UTC (14:00 and 15:00 local time) reach their maximum levels, regardless of material. However, only the high error observed at 16:00 UTC for opaque voxels is mitigated. Conversely, the error at 17:00 UTC (19:00 local time) results from underestimation, which, together with the one at 18:00 UTC (20:00 local time) , indicates that the abstraction level of the 3D tree representations is high given the sun's position during these hours. Also, the errors of cloudy days continue to increase steadily around solar noon.

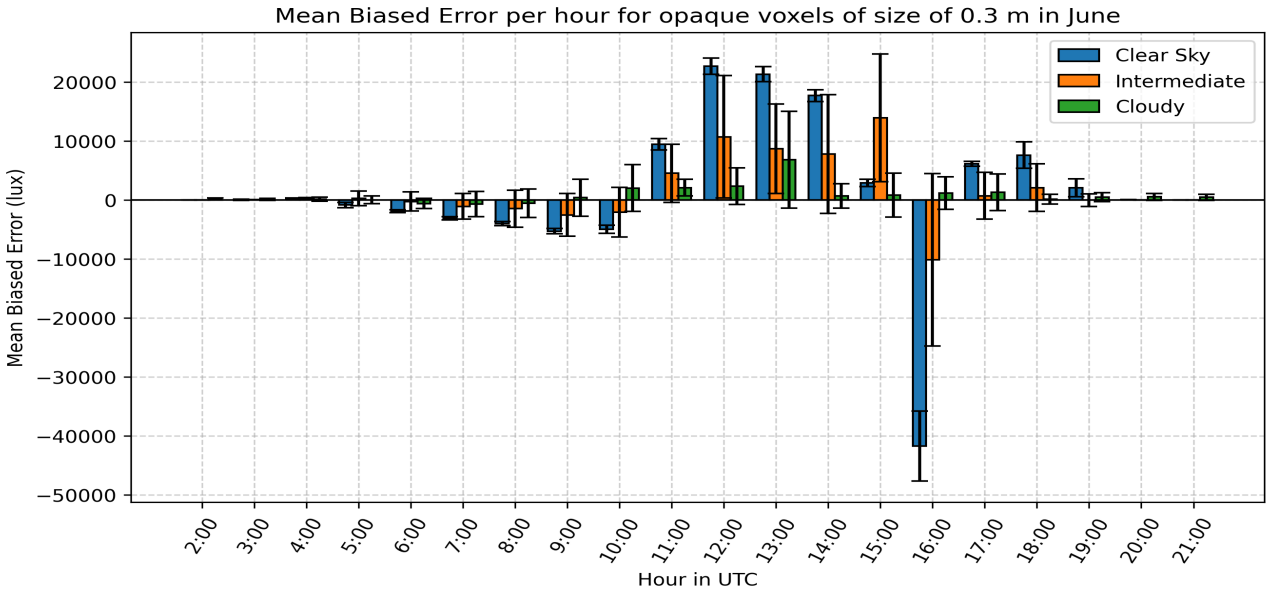


Figure 5.73.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.3 m in June under clear sky, cloudy, and intermediate conditions.

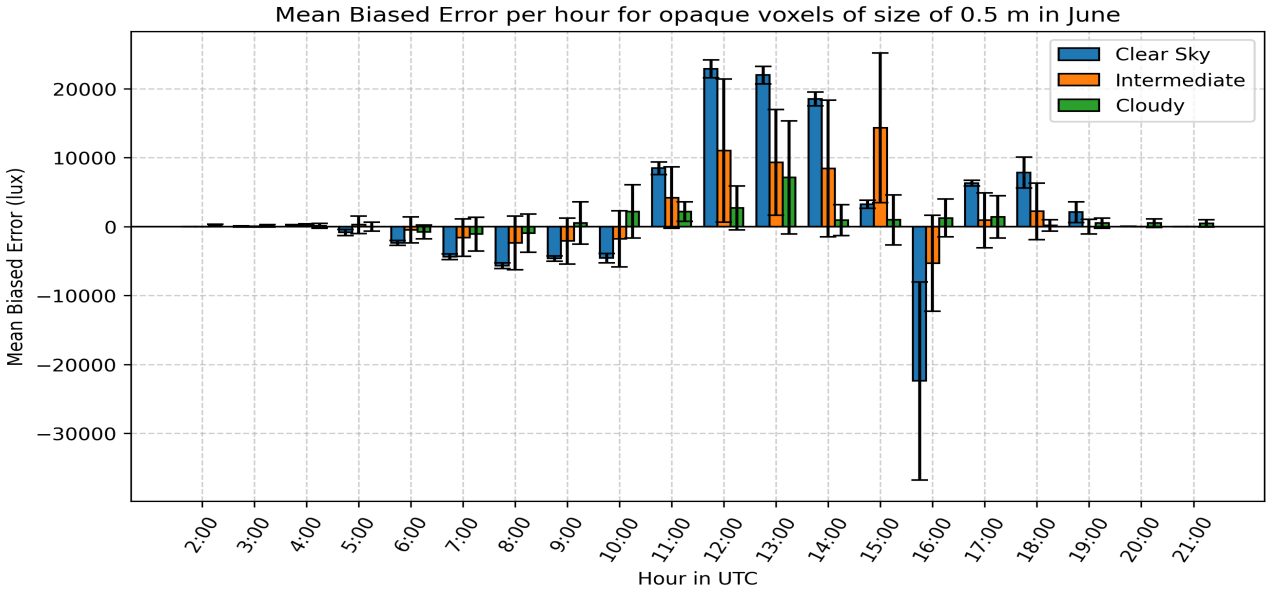


Figure 5.74.: Time series of Mean Bias Error with standard deviation per timestamp for opaque voxels of 0.5 m in June under clear sky, cloudy, and intermediate conditions.

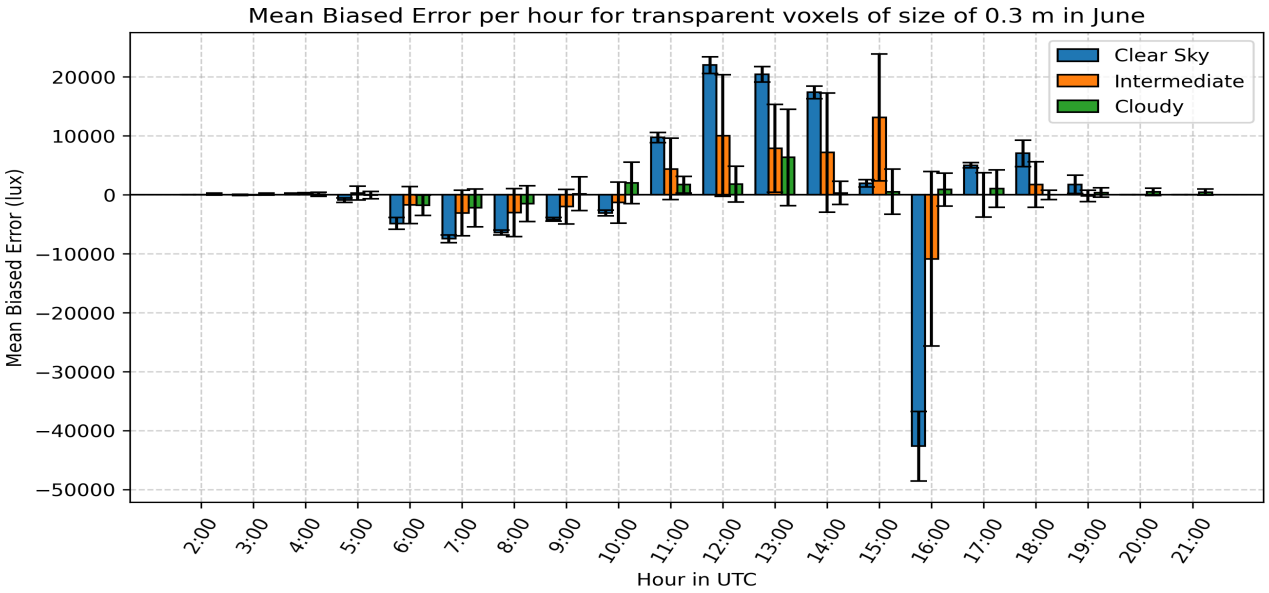


Figure 5.75.: Time series of of Mean Bias Error with standard deviation for transparent voxels of size of 0.3 m in June under clear sky, cloudy, and intermediate conditions.

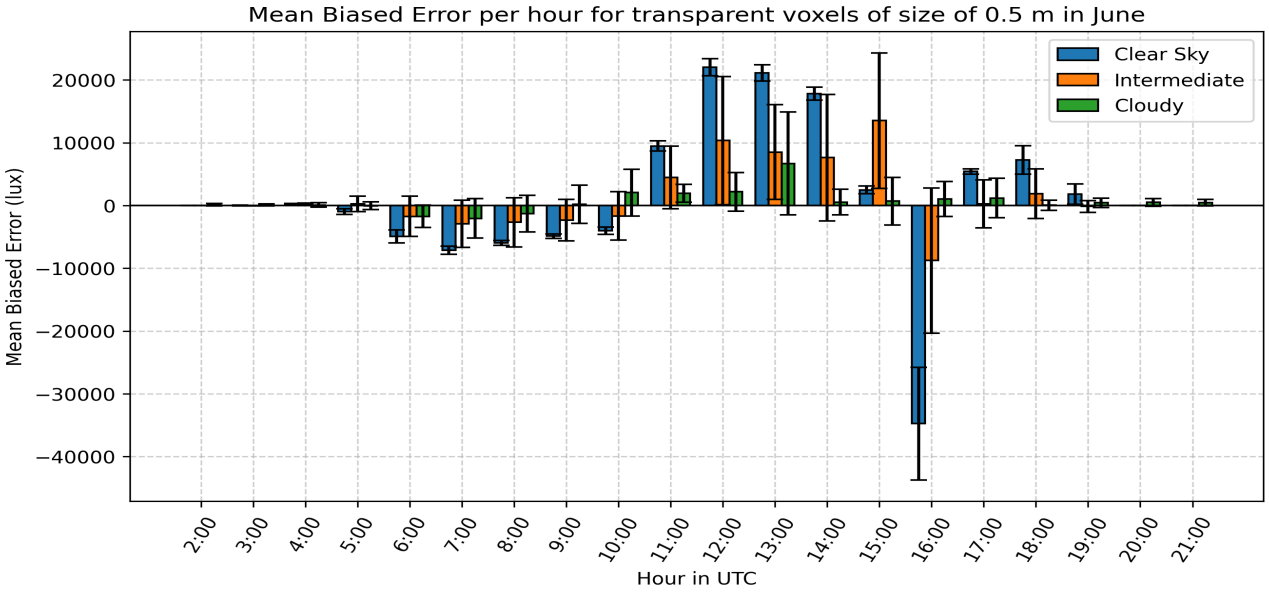


Figure 5.76.: Time series of of Mean Bias Error with standard deviation for transparent voxels of size of 0.5 m in June under clear sky, cloudy, and intermediate conditions.

Similar to the February analysis, the RMSE of the MBE values for all voxel cases was calculated to examine whether the voxel grid configurations that performed well for the overall monthly dataset also maintain consistent performance under different sky conditions. As it was already admitted, the June simulations were not as accurate as the February ones, potentially because of insufficient number of points and the sensitivity of the simulation for clear sky days. However, their result are still comparable with each other, since they were produced under the same circumstances (same scene and TMY file), and, therefore, there can be some conclusions.
For clear sky days, the tree 3D representations, that produced the smallest RMSE, were the voxel grid cases of transparent and opaque voxels of size of 0.5 m (11809.91 lux and 10335.17 lux respectively). This indicates that for clear sky days in June or in general months with more sunny days, research should be continued focusing on tree 3D representations, through the radiative activity in the simulation will be captured more accurately. Also, both for cloudy and intermediate sky conditions, the case of opaque voxels of sizes of 0.05 and 0.07 m, which aligns with some of the initial results, expect for the case of transparent voxels of size of 0.1 m.

Table 5.4.: RMSE of MBE values per voxel size and material under different sky conditions (in lux) in June.

				Clear sky		
Material	0.03 m	0.05 m	0.07 m	0.1 m	0.3 m	0.5 m
Transparent	14957.56	13047.18	13465.99	12740.00	12941.43	**11809.91**
Opaque	14285.57	12674.99	12528.77	12659.84	12845.66	**10335.17**
				Cloudy		
Transparent	2802.18	1608.36	2024.04	1525.17	1787.83	1873.84
Opaque	2610.71	**1418.71**	**1334.89**	1636.62	1822.08	1955.87
				Intermediate		
Transparent	6719.75	5060.05	5354.93	4929.79	5265.39	5231.38
Opaque	6279.36	**4846.23**	**4777.49**	5222.30	5417.51	5255.87

6. Conclusion

This section includes the conclusions the answers on the main research question and the sub-questions, based on the results obtained in this work. In addition, the limitations of the study are discussed, along with recommendations for future work to address them.

6.1. Research Overview

The sub-questions of this thesis were:

How can ALS point cloud data be used for tree 3D representation in urban daylight simulation?

ALS point cloud data can significantly enhance tree 3D representation in urban daylight simulations by capturing the geometric structure and form of trees. In regards to the former, trees can be represented by their point cloud data, a mesh representation such as alpha, primitive shape such as cylinders based on a skeletonization algorithm, polyhedra or voxel grids. However, specifically for the point cloud representation, often due to the low point density, derived from the characteristics of the scanning process or occlusion that is caused by the relative geometry between an object or a part of it and the aircraft, the complete representation of the tree is not facilitated. Thus, the other representations can cover this weakness with the reconstruction of surfaces, always based on the point cloud and occupy missing space. Specifically, about the voxel grid, it itself incorporate many ways to be represented in terms of the voxel size, orientation and shape of the voxels of the voxel grid.
Additionally, an important aspect of daylight simulation is understanding how these 3D tree representations interact with solar radiation. Point cloud can be helpful with this, it can estimate an object spectral attributes, either by using potential labels indicating that it belongs to a surface or spatial characteristics such as number of points enclosed in an object, number returns of these points, height of points and more.
In this research, as explained in Chapters 3 and 4 the cases of Point Cloud - based, Alpha Shape, Convex Hull and Voxel Grid were taken using AHN 5 point cloud data as input. Each case represented a different level of detail in the 3D tree representation: the Point Cloud–based case relied solely on the distribution of points, the Alpha Shape case modeled each tree as an individual mesh, and the Convex Hull case represented all tree branches collectively as a single, uniform solid object. On the other hand, about the Voxel Grid case, it was decided for this to be represented as a standard regular grid and to be assessed in terms of the voxel size and material, since these two parameters are still under investigation in academia.
These 3D representations were then incorporated into the scene on which the daylight simulation was performed. In addition, the fact that the whole methodology was based on open point cloud data, including the invaluable contribution of open source AdTree algorithm of Du et. al (2019) [22] and also yielded remarkable results especially for the February simulations, underscores the importance of the openness of the data.

What is the difference in accuracy of the results of the daylight simulation between diverse tree 3D representation approaches?

In both February and June, the simulation results show a similar pattern. The Point Cloud - based Case produced highly overestimated results, because the input point cloud lacked adequate density, resulting in a 'porous' tree 3D representation, due to the small cubes used to represent the points. On the other hand, the Alpha Shape and Convex Hull Case generate representations which are not easily permeable, since in the simulation they are assigned opaque materials leading to high level of underestimation of simulated illuminance values.
Regarding the Point Cloud - based Case, despite the simulation result of this research, this does not prove that the point cloud representation is not suitable for urban daylight simulation. A highly detailed tree point cloud can capture tree crown porosity, something that can be beneficial for this kind of applications. However, ALS point cloud data has limited point density and often lacks spatial information, especially in parts of the tree under the tree crown having an impact on trunk representation. In addition, it has been proven that Rhino software is not capable of handling point cloud data, and, therefore the implementation of this research was to approach a point cloud representation and approximate its radiative behavior as trees inside Rhino software.
Lastly, the results of the simulations of the Voxel Grid Case are diverse and dependent on the voxel size. After assessing the whole scale of the time series, it was concluded that voxel sizes close to 0.10 m can bring satisfactory results for February, yet the assigned material needs more investigation. In contrast, the Voxel Grid Case with sizes below 0.07 m tends to overestimate the simulated illuminance values, while voxel sizes larger than 0.10 m tend to underestimate them. The analysis between the Voxel Grid cases will be discussed in the next questions.

How can the results of daylight simulation be evaluated?

Overall, the assessment of these results is robust, as both the complete time series and the subsets of five days representing different sky conditions are examined to verify their consistency. First, the comparison of simulation results from different geometric representations was based on how well the simulated illuminance values matched the actual sensor measurements, using RMSE, MBE, as an initial evaluation. According to these metrics, Voxel Grid some cases with voxel sizes between 0.07 and 0.10 m yielded the most accurate daylight simulations, for February and June. Moreover, for each Voxel Grid Case the error distribution histogram was generated to see how errors behave across all of the time series and also additional metrics complemented the full time series results. For example, the nearest voxel distance distributions were used to examine how level of sparsity between the opaque voxel grids of different sizes, while for the second approach the distribution of voxel transmittance values across sizes was analyzed.
Then, the three separate dataset of clear sky, cloudy and intermediate conditions helped for highlighting which kind of days are more prone to simulation errors and how this impacts both the five-day and monthly sample. Also, it was proven that both in February and June the bigger underestimated errors occurred around the solar noon, when the direct light is stronger and there are no shades. Thus, this is another potential reason why the June simulations are not so accurate, together with the insufficient point density of the sized point cloud. Lastly, the RMSE of the MBE values of the hours assessed for each Voxel Grid Case was calculated to see the errors of which case approach more zero and, therefore, to define which cases of a particular voxel material and sky condition are the most accurate. After this, it was assessed whether the most accurate cases from the evaluation of the full time series align with the ones of each voxel material and sky condition.

What is the impact of seasonal alterations of the tree canopy in urban daylight simulation?

According to statistics shown at **Figures 5.1** and **5.39**, February cases achieved better overall performance, reflected in lower RMSE and MBE values. This indicates that the geometric representations, combined with the material assignments used for the June simulations, were less efficient in accurately modeling the trees than those applied in February. More specifically, during June the trees exhibit dense foliage, and consequently, a much higher number of points would be required to capture their structure accurately. These representations were based on synthesized points generated under certain limitations, the resulting point cloud contained an insufficient number of points to capture the tree conditions in June. Additionally, from the sky condition analysis, it was proven that on clear sky days the errors are larger, since inaccuracies in sky model and voxel geometry are more intense on these days. Thus, as June has more clear sky and longer days, the errors are more increased compared to February. Therefore, the influence of seasonal variation in the tree canopy on point cloud quality should be further investigated.

The main research question of this thesis was:

To what extent can the tree ALS point cloud data increase the accuracy of daylight simulations?

As discussed earlier, ALS point cloud data enables not only the creation of various types of 3D tree representations but also the estimation of their radiative properties. However, as demonstrated in this study, its relatively low point density limits the feasibility of detailed representations that more closely correspond to real-world tree structures. As a result, 3D representations such as the Point Cloud–based case did not perform as accurately as the others, as the spatial distribution of points was insufficient to represent the extent to which the trees obstruct sunlight reaching the CCC building.
Therefore, some abstractions of the geometric representation of the trees were exploited for this work. Regarding the Convex Hull Case, which demonstrates the biggest level of abstraction, since the branch points from all the trees where used to construct the convex hull, failed to represent accurately the trees and how they interact with light in the opposite direction compared to Point Cloud - based case, since the simulated illuminance values on the simulation surface are underestimated. However, not only the spatial representation, but also the assigned opaque materials contributed to this result, even though they were related to the trees of the study area (the wood material had high reflectance, reflecting on the light color of the trunks in the study area and the leaf material had a conventional reflectance value). Next, the tree 3D representation derived from the Alpha Shape Case, although it did not lead to as strong an underestimation of simulated illuminance values as the Convex Hull Case, since by representing individual trees it mitigates this effect, once again it did not manage to reflect on the trees' interaction, because these assigned materials proven to be unsuitable for this task.
On the other hand, the Voxel Grid Case provided more flexibility by creating more other one 3D representations, as the size could be changed. Generally, the voxel size for a voxel grid derived from ALS that should be used in these kinds of applications is still under investigation. The selection of diverse voxel sizes from smaller (0.03 m , 0.05 m and 0.07 m) to bigger (0.1 m, 0.3 m, 0.5 m) proved beneficial, providing valuable insights into how voxel resolution influences the simulation results. Additionally, combining voxel size with the assigned material parameter added further value to the research. Two material options were tested: in the first, opaque materials were applied, consistent with the previous cases and in the second, transparent materials were defined based on the point density within each voxel. Additionally, to estimate the behavior of the simulation in seasonal variation of the trees, ALS was enriched by additional points.
Based on Chapter 5, the results from February simulations, after combining the whole time series and sky conditions assessments, indicated that 3D voxel representations with sizes around 0.10 m are particularly effective, as they achieve a satisfactory balance between porosity and solidity. However, about the assigned material, it was not clear which of these two materials can contribute to better simulated values, since the results varied depending on the sky conditions. Therefore, the choice of voxel material requires further in-

vestigation to identify the optimal trade-off between light obstruction and transmittance. However, as it was already remarked, the June simulations are not as accurate as the February ones, probably because the point density of synthesized point cloud was not adequate to represent tree canopy and since the simulation system is susceptible to errors under clear-sky conditions. However, some conclusions can be drawn for future research. For cloudy and intermediate sky conditions, the cases of opaque voxels of sizes of between 0.05 and 0.07 m brought relatively satisfactory results, which match the whole time series assessment, and for clear sky condition days the case of voxel size of 0.5 m, regardless the material, perform in a better way, indicating that for simulation in months with many sunny days and longer days voxels of higher sizes are needed. Therefore, further investigation should be done for these kind of days.

6.2. Discussion and limitations

There are limitations which can be associated with to the AHN 5 point cloud data itself, data processing, the geometric representations and daylight simulation. Firstly, as discussed in Chapter 2, the product of the open AHN 5 point cloud is accompanied by errors resulting from the Light Detection and Ranging (LiDAR) acquisition method and from random inaccuracies in calculating precise position and time corrections between receivers and satellites. Therefore, these errors, arising from the scanning system itself, the occlusion, which prevents point collection in hidden areas, contribute to insufficient point density in the open point cloud data and affect the tree representation and, consequently, the simulation results. Also, the georeference of CCC Building also includes uncertainties, owing to the fact that this procedure was based on its LOD 2.2 model and rather than on corrections derived from in-situ survey measurements.
In addition, a part of the original AHN 5 point cloud was utilized for the February simulations, whereas the June simulations were based on a synthesized point cloud. This suggests that the February simulations are more reliable, which was not subjected to any modifications. Moreover, the synthesized point cloud was generated with only twice the density of the cleaned AHN 5 point cloud due to limitations in the capacity of the computer system used for this research. Using more sample points would potentially lead to better simulation results, especially for the Point Cloud Case and Voxel Grid Case of sizes below 0.07 m.
Regarding the geometric representations, methods similar to those of de Groot (2020) [32] and Du et al. (2019) [22] were not included in this work due to time constraints, as they would require additional processing of the individual tree point clouds. Such tasks could involve the calculation of tree parameters, as suggested by de Groot (2020) [32], while accounting for computational errors, as well as extending the framework of Du et al. (2019) [22] to ensure watertight tree models. Instead, the approaches adopted in this research ensured that the geometric representations were derived explicitly from the given point cloud and do not contain an additional error.
Additionally, about the Voxel Grid case, parameters like orientation and variable voxel size were outside of the scope of this study, because as explained in Chapter 3 it is likely that they would increase the computational time for constructing their 3D representation and also might not add value to the simulation results. On the other hand, the parameter of curvature of the voxels has not been studied in this work, since it would require more computational work, because an accurate 3D representation in this category should be built by the fusion of voxel grid and the convex hull. In regards to the Alpha Shape and Convex Hull Cases, the workflow is not fully automated: in the former, tree separation required manual refinement, and in the latter, points belonging to long branches touching the ground had to be removed manually after DBSCAN. Developing automated, efficient procedures for these steps would therefore be beneficial, especially in case the study area contains more trees.
Regarding daylight simulation, it has been proven that ClimateStudio plugin inserts its own errors in the simulation process by not handling clear sky conditions accurately. Potentially, this could be also attributed to simulation running setting. In case there was a configuration to increase the ambient ray samples, it is

possible that this would improve the Monte Carlo integration solution for the simulated illuminance. In addition, even if assigning the same materials on the other objects of the scene simulate the same conditions across all the cases, except for the transparent Voxel Grid cases, and, therefore, their results are comparable, the absence of exact real material definitions in Rhino also introduces errors into the simulations. Additionally, although in February the second approach of the Voxel Grid Case achieved a good fit of the simulated illuminance values at certain voxel sizes, the relationship between the glass material assigned to the voxels and the natural properties of trees, with or without vegetation, lies beyond the scope of this research.

6.3. Future Work

To address the limitations of this research, several recommendations can be taken into consideration. First, alternative methods of acquiring point cloud data could be explored, incorporating not only TLS data, but also UAV imagery, so that point clouds capturing branch structures in greater detail can be used in similar studies. Thus, for future studies, approaches similar to this introduced by indicated by Balestra et al. (2023) [8] for creating the dense point cloud by fusing many kinds of geospatial data and guidelines similar to those provided by Gassilloud et al. (2025) [28] and Maio et al. (2025) [44] for effective UAV preventing occlusion would be useful.
In addition, the processing regarding Alpha Shape and Convex Hull Cases should be improved in order to be become more user-independent. More specifically, for the former tree separation could be further improved by developing a more advanced graph structure to connect the points, similar to this proposed by Wang et al. (2021) [67] and for the latter the identification and exclusion of clusters belonging to long branches touching the ground could be facilitated by calculating of cluster features such as verticality, number of points, z-coordinate standard deviation, and others. Moreover, other more complex tree representations in different study areas could be assessed in terms of the accuracy of simulation results in a similar future work.
Furthermore, the study could be repeated using other software, such as Honeybee in order to evaluate the consistency of results across different simulation tools. Also, further investigation should be conducted regarding to the material that can be assigned to the voxels, which could potentially lead to the identification of more suitable options. Lastly, more research should be done regarding the behavior of the simulation not only among different months, but different sky conditions inside each month, so that the optimal 3D representation that simulates illuminance values in a better way under different sky conditions should be defined.

A. Reproducibility self-assessment

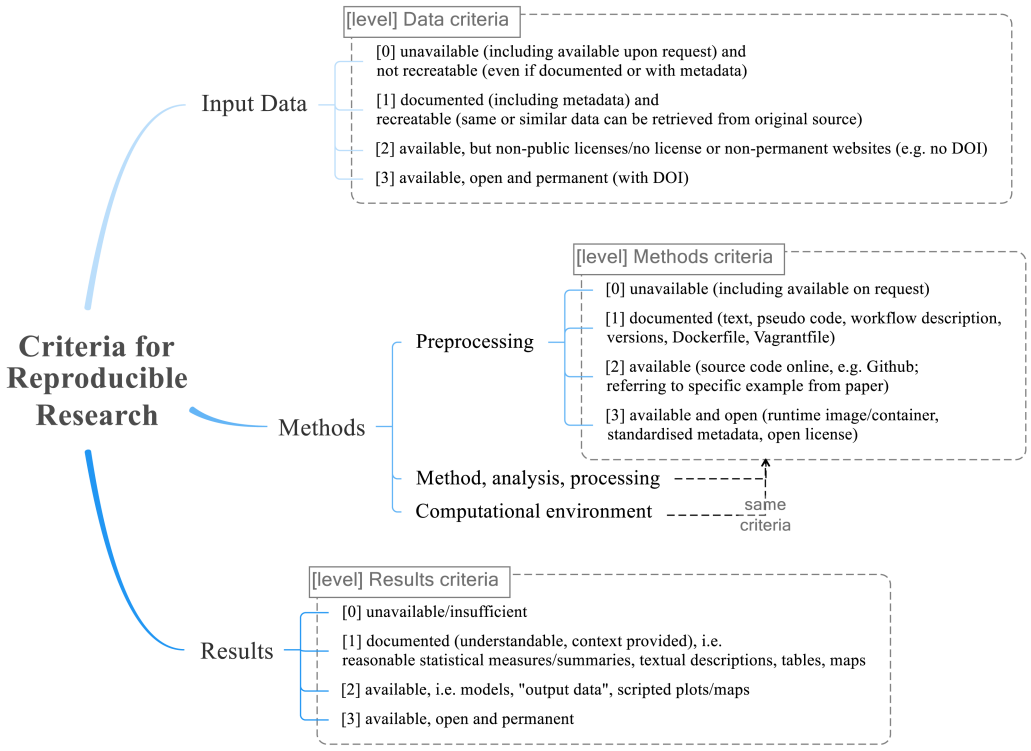


Figure A.1.: Reproducibility criteria to be assessed.

All executable files for the creation of the tree 3D representations are available in the GitHub repository associated with this thesis: https://github.com/viccic/Thesis_Victoria_Tsalapati

Table A.1.: Reproducibility criteria and their grades.

Criteria	Grade	Justification
Input data	1	`37EN2_16.LAZ` (available at https://geotiles.citg.tudelft.nl/)
	1	CAMS solar data (available at https://www.soda-pro.com/web-services/radiation/cams-radiation-service)
	0	CCC building 3D model and illuminance sensor data (available upon request)
Preprocessing	0	Code for cleaning AHN 5 is unavailable; however, it is provided as `37EN2_16_Trees.las`, together with the synthesized point cloud `Synthesized.las` and the point cloud containing leaf points for all trees `Leaves.las`.
Methods	2	Executables for the construction of each tree 3D representation and their explanation in README file are on Github.
Computational environment	1	Python 3.11
	0	Rhino software and ClimateStudio plugin (both require a license)
Results	2	Executables for extracting error distribution histogramms together with prints of RMSE and MBE values, nearest voxel distance distribution, time series of Mean Biased Error are on Github. Tree models with the simulation surface in 3dm Rhinoceros format are uploaded on 4TU repository.

Bibliography

[1] 3D BAG Team. *3D BAG Schema Documentation: Layers*. https://docs.3dbag.nl/en/schema/layers/. Accessed: 2025-09-15. 2025.

[2] 3DBAG. *3DBAG*. 2025. URL: https://3dbag.nl/en/viewer (visited on 03/20/2025).

[3] AHN (Actueel Hoogtebestand Nederland). *Dataroom*. 2025. URL: https://www.ahn.nl/dataroom (visited on 01/18/2025).

[4] Ryuhei Ando, Yuko Ozasa, and Wei Guo. "Robust Surface Reconstruction of Plant Leaves from 3D Point Clouds". In: *Plant Phenomics* 2021 (2021), p. 3184185. ISSN: 2643-6515. DOI: https://doi.org/10.34133/2021/3184185. URL: https://www.sciencedirect.com/science/article/pii/S264365152400075X.

[5] Ken Arroyo Ohori, Hugo Ledoux, and Ravi Peters. *3D Modelling of the Built Environment (v0.9.2)*. Lecture notes for GEO1004. 2024. URL: https://3d.bk.tudelft.nl/courses/geo1004/.

[6] Mohammed Ayoub. "100 Years of Daylighting: A Chronological Review of Daylight Prediction and Calculation Methods". In: *Solar Energy* 194 (2019), pp. 360–390. ISSN: 0038-092X. DOI: 10.1016/j.solener.2019.10.072.

[7] Priji Balakrishnan and J. Jakubiec. "Trees in Daylight Simulation – Measuring and Modelling Realistic Light Transmittance through Trees". In: *LEUKOS* 19 (Oct. 2022), pp. 1–28. DOI: 10.1080/15502724.2022.2112217.

[8] Mattia Balestra et al. "Geomatic data fusion for 3D tree modeling: The case study of monumental chestnut trees". In: *Remote Sensing* 15.8 (2023), p. 2197.

[9] Faizah Mohammed Bashir et al. "Effects of natural light on improving the lighting and energy efficiency of buildings: toward low energy consumption and CO2 emission". In: *International Journal of Low-Carbon Technologies* 19 (2024), pp. 296–305. DOI: 10.1093/ijlct/ctad130.

[10] Jon Louis Bentley, Franco P Preparata, and Mark G Faust. "Approximation algorithms for convex hulls". In: *Communications of the ACM* 25.1 (1982), pp. 64–68.

[11] D. E. Bertani et al. "Shedding Light on Light: A Review on the Effects on Mental Health of Exposure to Optical Radiation". In: *International Journal of Environmental Research and Public Health* 18.4 (2021), p. 1670. DOI: 10.3390/ijerph18041670.

[12] Brandywine Conservancy. *Invasive Species Spotlight: Norway Maple*. 2025. URL: https://www.brandywine.org/conservancy/blog/invasive-species-spotlight-norway-maple (visited on 01/18/2025).

[13] Daniela Buchalová et al. "Estimating Subcanopy Solar Radiation Using Point Clouds and GIS-Based Solar Radiation Models". In: *Remote Sensing* 17.2 (2025). ISSN: 2072-4292. DOI: 10.3390/rs17020328. URL: https://www.mdpi.com/2072-4292/17/2/328.

[14] Patrick Chazette et al. "5 - Principle and Physics of the LiDAR Measurement". In: *Optical Remote Sensing of Land Surface*. Ed. by Nicolas Baghdadi and Mehrez Zribi. Elsevier, 2016, pp. 201–247. ISBN: 978-1-78548-102-4. DOI: https://doi.org/10.1016/B978-1-78548-102-4.50005-3. URL: https://www.sciencedirect.com/science/article/pii/B9781785481024500053.

[15] Chire. *DBSCAN Illustration*. Accessed: 2025-09-08, licensed under CC BY-SA 3.0. 2011. URL: https://commons.wikimedia.org/wiki/File:DBSCAN-Illustration.svg.

[16] *Climate Studio Accuracy*. 2025. URL: https://www.solemma.com/blog/is-climatestudio-accurate (visited on 09/08/2025).

[17] *Climate Studio Speed*. 2025. URL: https://www.solemma.com/climatestudio/speed (visited on 09/08/2025).

[18] Codeway Apps. *PlantApp: Plant Identifier and Care Guide*. 2025. URL: https://play.google.com/store/apps/details?id=com.codeway.plantapp&pli=1 (visited on 01/18/2025).

[19] Copernicus Atmosphere Monitoring Service (CAMS). *CAMS Global and Regional Emission Inventories User Guide*. Tech. rep. Accessed: October 2025. European Centre for Medium-Range Weather Forecasts (ECMWF), 2021. URL: https://atmosphere.copernicus.eu/sites/default/files/2022-01/CAMS2_73_2021SC1_D3.2.1_2021_UserGuide_v1.pdf.

[20] Drury Crawley. "Creating Weather Files for Climate Change and Urbanization Impacts Analysis". In: Sept. 2007.

[21] Ian Dowman. "Integration of LiDAR and IfSAR for Mapping". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Istanbul, Turkey, July 12–23, 2004). Vol. XXXV.B2. ISPRS. 2004, pp. 90–100. URL: https://www.isprs.org/proceedings/xxxv/congress/comm2/papers/104.pdf.

[22] Shenglan Du et al. "AdTree: Accurate, detailed, and automatic modelling of laser-scanned trees". In: *Remote Sensing* 11.18 (2019), p. 2074.

[23] Frédo Durand. *Lecture 18: Ray Tracing*. MIT OpenCourseWare, Massachusetts Institute of Technology. Accessed: 30 September 2025. 2012. URL: https://ocw.mit.edu/courses/6-837-computer-graphics-fall-2012/resources/mit6_837f12_lec18/.

[24] Herbert Edelsbrunner and Ernst P. Mücke. "Three-dimensional alpha shapes". In: *ACM Trans. Graph.* 13.1 (Jan. 1994), pp. 43–72. ISSN: 0730-0301. DOI: 10.1145/174462.156635. URL: https://doi.org/10.1145/174462.156635.

[25] ersi. *Differential GPS Explained*. 2025. URL: https://www.esri.com/about/newsroom/arcuser/differential-gps-explained (visited on 09/01/2025).

[26] ersi. *Reader on GPS positioning*. 2025. URL: https://www.tudelft.nl/citg/over-faculteit/afdelingen/geoscience-remote-sensing/education/bsc-education/reader-on-gps-positioning (visited on 09/01/2025).

[27] Guangpeng Fan et al. "AdQSM: A new method for estimating above-ground biomass from TLS point clouds". In: *Remote Sensing* 12 (Sept. 2020), p. 3089. DOI: 10.3390/rs12183089.

[28] Matthias Gassilloud, Barbara Koch, and Anna Göritz. "Occlusion mapping reveals the impact of flight and sensing parameters on vertical forest structure exploration with cost-effective UAV based laser scanning". In: *International Journal of Applied Earth Observation and Geoinformation* 139 (2025), p. 104493. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2025.104493. URL: https://www.sciencedirect.com/science/article/pii/S1569843225001402.

[29] David Geisler-Moroder, Eleanor Lee, and Gregory Ward. "Validation of the Five-Phase Method for Simulating Complex Fenestration Systems with Radiance against Field Measurements". In: Aug. 2017. DOI: 10.26868/25222708.2017.401.

[30] Gemeente Delft. *Bomen in beheer door gemeente Delft*. 2025. URL: https://data.delft.nl/datasets/d83a50486b384bfe8038c2d762f5e628_0/explore?location=51.997184%2C4.379370%2C14.41 (visited on 01/18/2025).

[31] GeoTiles. *GeoTiles: High-Quality Tile Services for Mapping and Analysis*. 2025. URL: https://www.geotiles.nl/ (visited on 01/18/2025).

[32] Rob de Groot. "Automatic construction of 3D tree models in multiple levels of detail from airborne LiDAR data". PhD thesis. 2020.

[33] Erlend Hustad Honningdalsnes et al. "Benchmarking irradiation models for photovoltaic applications: A comparative analysis of radiance-based tools". In: *Solar Energy* 296 (2025), p. 113566. ISSN: 0038-092X. DOI: 10.1016/j.solener.2025.113566. URL: https://www.sciencedirect.com/science/article/pii/S0038092X25003299.

[34] Jozef Hrask. "Chronobiological aspects of green buildings daylighting". In: *Renewable Energy* 73 (2015), pp. 109–114. ISSN: 0960-1481. DOI: 10.1016/j.renene.2014.06.008.

[35] Tengping Jiang et al. "Segmentation of individual trees in urban MLS point clouds using a deep learning framework based on cylindrical convolution network". In: *International Journal of Applied Earth Observation and Geoinformation* 123 (2023), p. 103473. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2023.103473. URL: https://www.sciencedirect.com/science/article/pii/S1569843223002972.

[36] Benjamin Kahl. *Hardware Acceleration of Progressive Refinement Radiosity using Nvidia RTX*. Mar. 2023. DOI: 10.48550/arXiv.2303.14831.

[37] Abderrazzaq Kharroubi et al. "Three Dimensional Change Detection Using Point Clouds: A Review". In: *Geomatics* 2.4 (2022), pp. 457–485. ISSN: 2673-7418. DOI: 10.3390/geomatics2040025. URL: https://www.mdpi.com/2673-7418/2/4/25.

[38] Hugo Ledoux et al. *Computational modelling of terrains*. Version v2023.0. Delft University of Technology, 2023.

[39] Yuan Li et al. "SVDTree: Semantic Voxel Diffusion for Single Image Tree Reconstruction". In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 4692–4702. DOI: 10.1109/CVPR52733.2024.00449.

[40] lod. *lod 2.2*. 2025. URL: https://3dbag.nl/en/download?tid=9-288-552 (visited on 03/20/2025).

[41] Yuhao Lu et al. "Modeling the Shading Effect of Vancouver's Urban Tree Canopy in Relation to Neighborhood Variations". In: *Arboriculture & Urban Forestry (AUF)* 48.2 (2022), pp. 95–112. ISSN: 1935-5297. DOI: 10.48044/jauf.2022.008. eprint: https://auf.isa-arbor.com/content/48/2/95.full.pdf. URL: https://auf.isa-arbor.com/content/48/2/95.

[42] Andy McNeil. *The Five-Phase Method for Simulating Complex Fenestration with Radiance (Version 2.2)*. Tutorial notes. 2013. URL: http://www.radiance-online.org/learning/tutorials/fivephasetutorialfiles/test_5phase.bsh/at_download/file.

[43] Mecanoo Architects. *Co-Creation Centre and Nonohouse*. 2025. URL: https://www.mecanoo.nl/Projects/project/261/Co-Creation-Centre-and-Nonohouse (visited on 01/18/2025).

[44] Zhang Miao et al. "Drone LiDAR Occlusion Analysis and Simulation from Retrieved Pathways to Improve Ground Mapping of Forested Environments". In: *Drones* 9.2 (2025). ISSN: 2504-446X. DOI: 10.3390/drones9020135. URL: https://www.mdpi.com/2504-446X/9/2/135.

[45] N. Mie. "Master thesis". MA thesis. Aalborg University, 2023.

[46] Joseph B Murdoch. "Illumination engineering–from Edison's lamp to the laser". In: *(No Title)* (1985).

[47] Liangliang Nan. *Clustering*. Lecture notes, TU Delft, February 20, 2025. 2025.

[48] Pirouz Nourian et al. *Voxelization Algorithms for Geospatial Applications: Computational Methods for Voxelating Spatial Datasets of 3D City Models Containing 3D Surface, Curve and Point Data Models*. Lecture notes based on *MethodsX*, 3, 69–86. 2016. DOI: 10.1016/j.mex.2016.01.001.

[49] Fumio Okura. "3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping". In: *Breeding Science* 72.1 (2022), pp. 31–47. DOI: 10.1270/jsbbs.21074.

[50] PictureThis AI. *PictureThis: Identify Plants, Flowers, and Trees*. 2025. URL: https://www.picturethisai.com/ (visited on 01/18/2025).

[51] PlantNet Project. *PlantNet: A Citizen Science Project on Plant Biodiversity*. 2025. URL: https://plantnet.org/en/ (visited on 01/18/2025).

[52] Zhipeng Qu et al. "Fast radiative transfer parameterisation for assessing the surface solar irradiance: The Heliosat?4 method". In: *Meteorologische Zeitschrift* 26.1 (Feb. 2017), pp. 33–57. DOI: 10.1127/metz/2016/0781. URL: http://dx.doi.org/10.1127/metz/2016/0781.

[53] M. Schroedter-Homscheidt et al. "Surface solar irradiation retrieval from MSG/SEVIRI based on APOLLO Next Generation and HELIOSAT?4 methods". In: *Meteorologische Zeitschrift* 31.6 (Dec. 2022), pp. 455–476. DOI: 10.1127/metz/2022/1132. URL: http://dx.doi.org/10.1127/metz/2022/1132.

[54] Scratchapixel 2.0. *Light Transport and Ray-Tracing: Whitted Ray-Tracing*. Accessed: 30 September 2025. 2023. URL: https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-overview/light-transport-ray-tracing-whitted.html.

[55] SoDa Service. *SoDa: Solar Data and Tools for Professionals*. 2025. URL: https://www.soda-pro.com/home (visited on 01/18/2025).

[56] Sarith Subramaniam. *Daylighting Simulations with Radiance using Matrix-Based Methods*. https://www.radiance-online.org/learning/tutorials/. Accessed: 2025-10-16. Oct. 2017.

[57] Michal Szucki and J.s Suchy. "A voxelization based mesh generation algorithm for numerical models used in foundry engineering". In: *Metallurgy and Foundry Engineering* 38 (Jan. 2012), p. 43. DOI: 10.7494/mafe.2012.38.1.43.

[58] Tech Soft 3D. *Global Illumination*. HOOPS Luminate 2024.7.0 Documentation. 2024. URL: https://docs.techsoft3d.com/luminate/latest/book/subjects/bk_re/bk_re_srd/bk_re_sw3d_gi.html.

[59] The Green Village. *Public Dashboard: Weather Station Data*. https://thegreenvillage.grafana.net/public-dashboards/617e43ba8b2f4a86952e11dff479f8cd?orgId=1. Accessed: October 16, 2025. n.d.

[60] Panagiota Theodoropoulou. *Optimized control strategy for Venetian blinds in an event space with fully glazed facades*. 2023.

[61] B. Tian, R.C.G.M. Loonen, and J.L.M. Hensen. "Combining point cloud and surface methods for modeling partial shading impacts of trees on urban solar irradiance". In: *Energy and Buildings* 298 (2023), p. 113420. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2023.113420. URL: https://www.sciencedirect.com/science/article/pii/S0378778823006503.

[62] Time and Date AS. *Sunrise and sunset times for Delft, February 2023*. Accessed: October 12, 2025. 2023. URL: https://www.timeanddate.com/sun/@51.99655,4.37668?month=2&year=2023.

[63] Lorie Toma. *Computational Geometry: Convex Hull in 3D*. Lecture slides. [Accessed: 7-September-2025]. 2017. URL: https://tildesites.bowdoin.edu/~ltoma/teaching/cs3250-CompGeom/spring17/Lectures/cg-hull3d.pdf.

[64] Peter Tregenza and I. Waters. "Daylight coefficients". In: *Lighting Research Technology - LIGHTING RES TECHNOL* 15 (June 1983), pp. 65–71. DOI: 10.1177/096032718301500201.

[65] J. Trochta et al. "3D Forest: An application for descriptions of three-dimensional forest structures using terrestrial LiDAR". In: *PLoS ONE* 12.5 (2017), e0176871. DOI: 10.1371/journal.pone.0176871. URL: https://doi.org/10.1371/journal.pone.0176871.

[66] Vossman and M. W. Toews. *A set of voxels in a stack, with a single voxel shaded*. Licensed under CC BY-SA 2.5; accessed 7-September-2025. 2007. URL: https://commons.wikimedia.org/wiki/File:Voxelisation.svg.

[67] Di Wang, Xi Liang, Gilbert I. Mofack, et al. "Individual tree extraction from terrestrial laser scanning data via graph pathing". In: *Forest Ecosystems* 8.1 (2021), p. 67. DOI: 10.1186/s40663-021-00340-w. URL: https://doi.org/10.1186/s40663-021-00340-w.

[68] Jinhu Wang. *TreeSeparation: Individual tree segmentation from LiDAR point clouds*. https://github.com/Jinhu-Wang/TreeSeparation. Accessed: 2025-09-15. 2020.

[69] Hannah Weiser et al. "Opaque voxel-based tree models for virtual laser scanning in forestry applications". In: *Remote Sensing of Environment* 265 (2021), p. 112641. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2021.112641. URL: https://www.sciencedirect.com/science/article/pii/S0034425721003618.

[70] Wikipedia Contributors. *Acer Platanoides*. 2025. URL: https://en.wikipedia.org/wiki/Acer_platanoides (visited on 01/18/2025).

[71] Wikipedia Contributors. *Illuminance*. 2025. URL: https://en.wikipedia.org/wiki/Illuminance (visited on 01/18/2025).

[72] Wikipedia contributors. *Voxel — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-September-2025]. 2025. URL: https://en.wikipedia.org/wiki/Voxel.

[73] Han Xu et al. "3D Tree Reconstruction in Support of Urban Microclimate Simulation: A Comprehensive Literature Review". In: *Buildings* 11.9 (2021). ISSN: 2075-5309. DOI: 10.3390/buildings11090417. URL: https://www.mdpi.com/2075-5309/11/9/417.

[74] Yusheng Xu, Xiaohua Tong, and Uwe Stilla. "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry". In: *Automation in Construction* 126 (2021), p. 103675. ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2021.103675. URL: https://www.sciencedirect.com/science/article/pii/S0926580521001266.

[75] Zhipeng Zeng et al. "A Bottom-Up Multi-Feature Fusion Algorithm for Individual Tree Segmentation in Dense Rubber Tree Plantations Using Unmanned Aerial Vehicle–Light Detecting and Ranging". In: *Plants* 14.11 (2025). ISSN: 2223-7747. URL: https://www.mdpi.com/2223-7747/14/11/1640.

Colophon

This document was typeset using LaTeX, using a modified the KOMA-Script class `scrbook` available at https://github.com/tudelft3d/msc_geomatics_thesis_template. The main font is Palatino.