Now you see me, now you don't

Direct LiDAR-Based Intervisibility Modeling with Semi-Transparent Vegetation for Military Mission Planning

Yair Roorda

June 2026

A thesis submitted to the Delft University of Technology for the degree of MSc Geomatics

Yair Roorda: *Now you see me, now you don't* (2026)

This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

This document was typeset using Typst, and uses the template freely available at https://github.com/tudelft3d/msc_geomatics_thesis_typst.

The work in this thesis was carried out in cooperation with:



Geo-Database Management Centre
Delft University of Technology



Data Science Center of Excellence

Supervisors:	Prof.dr. Peter van Oosterom
	Dr. Martijn Meijers
	Ir. Nick Leenders
Co-reader:	Dr. Felix Dahle

Preface

This report is a result of my thesis *Direct LiDAR-Based Intervisibility Modeling with Semi-Transparent Vegetation for Military Mission Planning* at the Delft University of Technology for the degree of MSc Geomatics.

I am grateful to my supervisors Peter van Oosterom, Martijn Meijers, and Nick Leenders for their guidance and feedback over the past months. As well as my co-reader, Felix Dahle, and all other faculty staff for their time and energy.

This work was done in close collaboration between the GDMC research group of the TU Delft and the DSCE of the Ministry of Defense. I would like to sincerely thank my colleagues and fellow students at these institutes for their flexibility, enthusiasm, and expertise.

My thanks also go out to my fellow students for their feedback and community. Finally, I want to thank my friends and family for their patience while listening to me ramble on about this thesis and their support during the process.

Contents

Preface ..... **i**

Glossary **1**

Summary **2**

1 Introduction **3**
1.1 Topic and motivation 3
1.2 Scope 3
1.3 Research questions 4
1.4 Supervisory team 4
1.5 Readers guide 5

2 Paper **6**
2.1 Introduction 7
2.2 Related work 8
2.3 Methodology 9
2.4 Results 14
2.5 Discussion 19
2.6 Conclusion 20

3 Background **22**
3.1 Airborne Mission planning 22
3.2 Visibility analysis 23
3.3 Laser scanning 24
3.4 Data 26
3.5 COPC 29
3.6 Point cloud processing libraries 30
3.7 Segmenting vegetation in point clouds: 31
3.8 Beer-Lambert attenuation law 32
3.9 Literature review 32

4 Reflection **40**
4.1 Extended discussion 40
4.2 Answers to research questions 47
4.3 Main contributions 48
4.4 Design process 48
4.5 Goals 49

Bibliography **52**

A Reproducibility self-assessment **57**

B Declaration of LLM usage **58**

C Keyword queries **59**

Glossary

Term	Definition
2.5 D	A model where each location(x,y) is assigned exactly one height(z)
AHN	General height model of the Netherlands
ALS	Airborne Laser Scanning
AOI	Area of Interest
CHM	Canopy Height Model
COPC	Cloud Optimized Point Cloud
DEM	Digital Elevation Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
FDR	False Detection Rate
fuzzy viewshed	A model in which each point is assigned a graded value between fully visible and fully occluded
HAG	Height Above Ground
HPR	Hidden Point Removal
intervisibility	Asks whether a certain target is visible to a certain source and vice-versa
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LoS	Line of Sight
NIR	Near-Infrared
non-binary viewshed	Considers varying degrees of visibility rather than a binary visible/invisible model
probabilistic viewshed	A model in which each point is assigned a probability that the target is visible
SfM	Structure from Motion. Photogrammetric technique for estimating 3D scenes from images
TIN	Triangulated Height Model
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle

Summary

In modern warfare, precise visibility calculations are critical to the success of airborne operations, such as those involving drones and helicopters. Currently, military mission planning remains a largely manual process. Furthermore, existing automated methods for calculating line-of-sight (LoS) primarily rely on simplified 2.5D representations, such as grids or voxels. Most methods either ignore the visual impact of vegetation entirely or unrealistically treat it as a fully opaque obstacle. To support safer, faster, and more standardized data-driven operations, this thesis proposes a novel method for calculating fully 3D, non-binary intervisibility directly from airborne LiDAR point clouds.

The proposed pipeline is designed to efficiently handle large-scale LiDAR point cloud data by leveraging Cloud Optimized Point Cloud (COPC) streaming alongside k-d tree spatial indexing. To enhance the raw LiDAR data, the methodology uses the deep learning framework Myria3D to automatically segment vegetation. Additionally, a preprocessing step synthesizes points along vertical building facades to correct for the inherent angle limitations of airborne LiDAR scanning.

A core innovation of this research is the ability to model vegetation as a semi-transparent obstacle rather than a solid barrier. The algorithm evaluates visibility along a discrete, volumetric LoS cylinder. Instead of yielding a traditional binary "visible" or "not visible" output, it calculates a continuous probability score. This is achieved by adapting the Beer-Lambert law, using the local density of LiDAR vegetation points to estimate permeability and apply exponential visual attenuation. For optimal computational performance, the algorithm includes an early-exit mechanism that immediately terminates the calculation if the LoS intersects a solid obstacle, such as a building or the terrain.

By aggregating these individual LoS calculations, the method produces 2D, 3D, and cumulative viewsheds representing multi-observer environments. The resulting data can be visualized directly as 3D point clouds, voxelized volumes, web-based models, or 2D map overlays. To demonstrate the method's practical utility in military planning, it is applied to three operational prototypes: safe flight height mapping, visibility-aware routing, and optimal lookout site selection.

Ultimately, this direct point cloud analysis bridges the gap between raw spatial data and high-level decision-making. By realistically capturing partial occlusions, the model effectively increases the recognized hidden areas in which missions can operate and grants analysts the flexibility to implement precise, threshold-based risk management. Future research should focus on empirically determining specific attenuation constants, more accurately modeling the human field of view, and applying the algorithm to alternative data sources like Structure from Motion (SfM) point clouds.

1 Introduction

This chapter introduces the core motivation, objectives, and structure of this thesis. It begins by establishing the context and relevance of the topic. Next, it defines the scope of the research. Then, the primary and secondary research questions that guide the study are presented. Finally, the section introduces the supervisory team and concludes with a reading guide for the remainder of the thesis.

1.1 Topic and motivation

With tensions rising globally and in Europe, especially since the start of the Russo-Ukrainian War, revitalising the Dutch military will be an important challenge over the coming years. A key area of investment is the use of data science and data-driven decision making in every branch of the military (Ministerie van Defensie, 2024, p. 31-32).

Currently, military mission planning is a largely manual process (Maaiveld et al., 2024, p. 2-3). Fitting the global trend, the Dutch military is increasingly implementing data-driven analysis and techniques across the organization (Ministerie van Defensie, 2024). Doing the same for airborne mission planning would support safer, faster, more standardized, and cheaper operations.

This thesis investigated the use of direct LiDAR point cloud analysis for data-driven planning of helicopter and drone missions. It proposes a novel method for calculating line of sight with probabilities as described in Section 2.3. It improves current methods in four key ways: moving from simplified 2.5D to true 3D representations; performing analysis directly on point clouds; taking into account vegetation; and modeling vegetation as a semi-transparent obstacle. The last of which is done by approximating the permeability of the vegetation using point density and an exponential attenuation model.

This method is then applied to practical problems related to military airborne mission planning as presented in Section 2.4, bridging the gap between the current 2D analog methods and 3D data-driven methods.

The new model is evaluated both qualitatively and quantitatively by comparing it to existing workflows. This includes a comparison to conventional GIS methods using the QGIS software (Inglis et al., 2022). Additionally, the impact of representing trees, seasonal differences, and transparency is evaluated. Finally, the scalability and performance of the point cloud-based method are assessed.

1.2 Scope

Military mission planning requires modeling many independent and sometimes highly dynamic effects. Leenders et al. (2025) provides a systematic overview of these different effects. Effects that influence the planning decisions include, but are not limited to: acoustics, background camouflage, solar glare, safe landing zones, radar detection, and egress routes.

Each of these aspects is interesting and deserving of research in its own right. However, this thesis focuses on the inclusion of vegetation in non-binary visibility as the most pressing and influential factor for current planning processes.

The goal of this thesis is to develop a tool to assist mission planning, meaning before the operation is underway. This means there is no requirement for the tool to run in real-time or on limited onboard hardware. However, computational performance remains a priority and a major bottleneck of current methods. As LiDAR produces large datasets, runtime of individual Line of Sight (LoS) calculations must be highly optimized. This is a requirement to run higher-level analyses like cumulative viewsheds from multiple observers in a reasonable time.

The method relies exclusively on LiDAR data and does not directly include other sources like satellite imagery, photogrammetry, or radar. The main reasons for this are simplicity, flexibility of the method, and feasibility constraints.

While the research may apply to similar problems like optimizing airborne mission routing or others in fields like ecology, archaeology, and urban planning, only military mission planning is considered for the design decisions made in this thesis.

No field data will be collected for validation, nor will additional data be collected. While crucial for many practical applications, it is out of scope of this thesis for feasibility reasons.

Finally, while there is overlap, the method is designed to mimic human vision and not that of radar, camera, or infrared imaging.

1.3 Research questions

The main research question that is investigated is:

> *How can military aerial mission planning be improved by incorporating semi-transparent vegetation in probability-based visibility calculations computed directly on airborne LiDAR point clouds?*

This main question is further divided into six sub-questions:

1. *How can point clouds be automatically and efficiently accessed for performing LoS visibility queries using available frameworks?*
2. *How can vegetation be automatically classified in point clouds using known methods?*
3. *How can permeability parameters be derived from point-cloud features so that they fit within an exponential attenuation model of visibility?*
4. *How can 3D LoS calculation best be visualized for use in military airborne mission planning?*
5. *How does incorporating semi-transparent vegetation influence the accuracy and performance of visibility calculations?*
6. *What is the accuracy of the proposed method compared to existing methods and compared to reality?*

1.4 Supervisory team

This thesis was supervised by two different institutions, as discussed below. In this collaboration, the GDMC group of the TU Delft was mainly responsible for academic and scientific rigour. The DSCE was mainly responsible for providing subject-specific expert input on the military application.

Geo-Database Management Center

The GDMC[1] is a research and development center for all activities related to the modeling, storage, retrieval, analysis, presentation, and distribution of geo-information. The GDMC lab

[1] https://www.gdmc.nl/

is part of the Architectural & Geo Informatics Research Group of the Faculty of Architecture and the Built Environment, TU Delft.

The GDMC carries out fundamental research on the use of spatial data types, operators, functions, clustering, and indexing in database management systems. One of the research topics within this theme is large point cloud datasets.

Data Science Center of Excellence

The DSCE[2] is the center of expertise within the Dutch Ministry of Defense in the field of data science & AI. It is the result of a collaboration between the Faculty of Military Sciences from Tilburg University and the Ministry of Defense. Additionally, the DSCE collaborates with external knowledge institutes like the Netherlands Aerospace Center[3] (NLR) in this case.

Its mission is to improve the effectiveness of the armed forces through better military decision-making by developing and sharing data science and AI knowledge.

The DCSE achieves this by conducting scientific research, developing and providing education, and offering advice to partners. The DSCE is organized according to five themes vital to defense, as outlined in the Defense strategy for data science and AI (Ministerie van Defensie, 2023). This thesis falls under the theme: military decision-making and intelligence.

1.5 Readers guide

The thesis continues with Chapter 2, which contains the main body of the work presented as a stand-alone paper with the intent of submitting to ISPRS journal[4] aimed at subject-matter experts. Chapter 3, provides the necessary background information and definitions aimed at a general geomatics or engineering audience. It covers the basic definition of both visibility analysis and military applications, as well as an extended discussion of existing literature. Finally, Chapter 4 contains an extended discussion and reflection on both the material and process of the thesis.

[2] https://faculteitmilitairewetenschappen.nl/page/view/e6fc6085-7b66-4aa8-a981-b3da276d7a82/data-science-centre-of-excellence-dsce

[3] https://www.nlr.org/nl/

[4] https://www.sciencedirect.com/journal/isprs-journal-of-photogrammetry-and-remote-sensing

2 Paper

Page intentionally left blank.

Direct LiDAR-Based Intervisibility Modeling with Semi-Transparent Vegetation for Military Mission Planning

Yair Roorda[1], Peter van Oosterom[2], Martijn Meijers[2], Nick Leenders[3]

[1] TU Delft
[2] TU Delft
[3] NLR

Keywords: Probabilistic visibility, Military mission planning, Direct point cloud processing, Vegetation aware visibility.

Abstract

In modern warfare and airborne operations, such as those carried out by drones and helicopters, precise visibility calculations are critical to mission success. However, current automated methods for calculating line-of-sight (LoS) often rely on simplified gridded, meshed, or voxelized 2.5D representations and either ignore vegetation's impact or unrealistically model it as fully opaque. To address these limitations, this research proposes a novel method for calculating fully 3D, non-binary intervisibility directly from airborne LiDAR point clouds. The method leverages Cloud Optimized Point Cloud (COPC) streaming and k-d tree spatial indexing to rapidly query large, remote datasets. Vegetation is automatically segmented using the deep learning Myria3D framework. The algorithm computes viewsheds by modeling terrain, buildings, and semi-transparent vegetation, resulting in a continuous probability score rather than a traditional binary output. This is achieved by adapting the Beer-Lambert law to apply exponential attenuation along a discrete, volumetric LoS. Finally, the results are visualized as 3D spatial volumes or 2D map overlays for practical integration into mission planning workflows, such as safe flight height mapping and visibility-aware routing. This direct point cloud analysis bridges the gap between raw data and high-level spatial decision-making, offering a framework for safer, data-driven mission planning.

1. Introduction

Tensions are rising globally and especially in Europe after the start of the Russo-Ukrainian War. Member states' defense spending has increased 31% since 2021, reaching 1.9% of the EU's combined GDP or € 326 billion in 2024 (European Commission, 2025, p. 16)

Like many allies, revitalizing the Dutch military will be an important challenge over the coming years. This priority is emphasized by increased spending to € 24 billion annually and a new personnel inflow target of at least 9,000 per year from 2029 (Ministerie van Defensie, 2024, p. 6, 44).

A key area of investment is the use of data science and data-driven decision-making in every branch of the military (Ministerie van Defensie, 2024, p. 31-32). One area where much progress can be made is in the data-driven planning of airborne missions. With the explosive growth of drones (Granelli et al., 2026) and electronic warfare (Reddy & Sinha, 2025), future systems are likely to become even more reliant on data-driven planning. This is especially relevant for autonomous operations, which have been and will likely continue to be a growing trend (Gettinger, 2020). This goes for military tasks like surveillance (Govindaraju et al., 2014) but also search and rescue applications.

Visibility analysis is a highly multidisciplinary and growing field (Inglis et al., 2022). Algorithms and techniques have been applied to many different domains, like: optimal placement of lookouts for firefighters (Mistick et al., 2022), maximization of cell tower range (Kashyap et al., 2014), archaeology (Alblas, 2012; Llobera, 2007; Wheatley, 1995), advertisement (Chmielewski, 2021), optimal placement of surveillance cameras (Yaagoubi et al., 2015), ecology (Gresham et al., 2023; Zong et al., 2023), GNSS satellite visibility (Gheorghiu, 2024; Lu & Han, 2020), and urban planning (Bartie et al., 2010; Peters et al., 2015; Zhang et al., 2021).

Visibility analysis is also an important aspect during the planning of airborne operations. Analysis of the line of sight (LoS) between airborne vehicles like UAVs and helicopters is a crucial source of information for optimizing the effectiveness and safety of military equipment and personnel.

Currently, military mission planning is a largely manual process (Maaiveld et al., 2024, p. 2-3). Fitting the global trend, the Dutch military is increasingly implementing data-driven analysis and techniques across the organization (Ministerie van Defensie, 2024). Doing the same for airborne mission planning would support safer, faster, more standardized, and cheaper operations.

Available automated methods for estimating line of sight fall short in four key ways: vegetation, non-binary visibility, dimensionality, and representation. These are discussed in detail in Section 2.2, which reviews existing literature. It also discusses improvements for each of the four shortcomings in the 3D context or airborne mission planning.

The main research question investigated in this paper is: *How can military aerial mission planning be improved by incorporating semi-transparent vegetation in probability-based visibility calculations computed directly on airborne LiDAR point clouds?*

This paper proposes a new method for performing visibility analysis directly on 3D point clouds. Section 2.3 details the proposed method and the core five-stage pipeline for direct point cloud processing, including the generation of spatial products and their practical application to military airborne mission planning. Section 2.4 presents the results of the research as well as an evaluation of the new model, assessing its accuracy and performance. Finally, Section 2.5 explores the implications and limitations of the proposed method, while Section 2.6 summarizes the core findings and directions for future work.

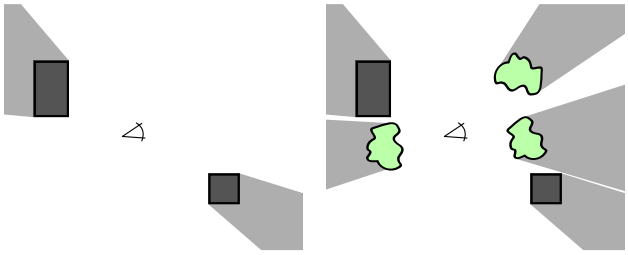
2. Related work

2.1 Vegetation

Vegetation is often neglected in visibility analysis (Oude Elberink, 2010, p. 6; Peters et al., 2015; Rasova, 2018; Zhao et al., 2024). This is because, in most models, vegetation and specifically trees are significantly harder to represent than terrain or building geometry.

Still, vegetation plays a significant role in visibility in both urban and rural areas (Alblas, 2012; Zhang et al., 2025; 2021). Pyysalo et al. (2009) shows that including trees in visibility analysis for a forested test site in Finland can reduce visible volume by as much as 53%.

Figure 1 shows how including vegetation in a toy example could significantly reduce visibility.



(a) (b)

Figure 1: Diagram showing the difference between a viewshed that does not **(a)** and does **(b)** include vegetation.

There has been consistent progress in taking into account vegetation in visibility analysis. Guth (2009) provides a key starting point. Their method achieves this using two raster grids, a DTM and a DSM, derived from LiDAR measurements. This early implementation approximates the included vegetation as fully opaque.

Bartie et al. (2011) includes vegetation using a separate raster called a canopy height model or CHM. Their results indicate that including vegetation makes the analysis both more stable and more accurate. They also include a binary mask for deciduous and evergreen trees, where deciduous trees are set to a single minimal visual permeability during winter.

Murgoitio et al. (2013) include trees by focusing only on tree trunks, excluding leaves or branches. They use a raster as an intermediate representation and calculate full viewsheds using the ArcGIS software.

Applying these concepts to mission planning, Govindaraju et al. (2014) incorporate vegetation into a probabilistic visibility model for UAV surveillance path planning in cluttered environments. They approximate partial occlusion from tree canopies by assuming a constant mean vegetation height and applying the Beer-Lambert attenuation law based on tree cover density.

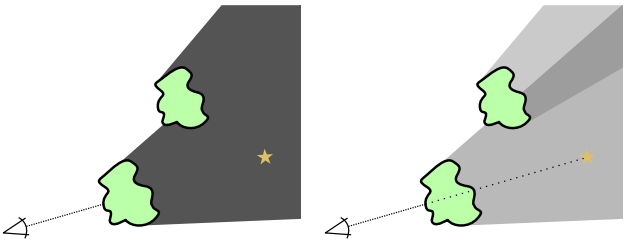
More recently, Maaiveld et al. (2024) offer an implementation including vegetation in a non-binary visibility algorithm specific to military mission planning. They apply their method to autonomous ground vehicles and path optimization. They first convert the raw point clouds to a grid, in this case hexagonal. Their model for attenuation approximates the effect as a linear combination using fixed obstruction weights for trees and bushes.

2.2 Non-binary visibility

Non-binary visibility is consistently mentioned as a limitation and topic for future work in past and current literature (Fisher-Gewirtzman et al., 2013; Pyysalo et al., 2009; Zhang et al., 2021; Zhao et al., 2024). Maps that display viewshed are almost always binary, meaning only ‘visible’ and ‘not visible’ are represented. However, in reality, visibility is a continuous scale. Vegetation in particular is almost always semi-transparent. Most methods represent it as completely opaque, causing visibility to be underestimated around and behind vegetation.

In the rare cases that transparency is considered, the attenuation is often approximated by relying on some linear metric like, for example, the minimum (Maaiveld et al., 2024). While practical for calculation, it fails to accurately represent reality when compared to exponential models like those derived from the Beer-Lambert attenuation law (Dean, 1997; Govindaraju et al., 2014; Llobera, 2007).

Figure 2 shows how modeling vegetation as semi-transparent allows for visibility to be partially obstructed.



(a) (b)

Figure 2: Diagram showing the difference between a binary viewshed **(a)** and a probabilistic viewshed **(b)** of visibility.

The first form of non-binary visibility was introduced to the GIS domain by Fisher (1994), who modeled it as the uncertainty in a viewshed arising from Digital Elevation Model (DEM) errors.

Since then, the concept has expanded. Recently, Leenders et al. (2025) mapped the broader evolution of non-binary visibility in their scoping literature review. They provide a systematic overview of the various factors that introduce uncertainty into viewsheds, specifically identifying vegetation as a primary driver and mentioning military helicopter mission planning as a key application for these techniques.

While researchers have long recognized that vegetation acts as a semi-transparent medium rather than a solid wall, practically computing this non-binary visibility remains a consistent challenge and a recurring topic for future work in spatial analysis literature (Fisher-Gewirtzman et al., 2013; Pyysalo et al., 2009; Zhang et al., 2021; Zhao et al., 2024).

Dean (1997) establishes the first method that models visibility where vegetation is represented as a semi-transparent obstacle. Dean identifies 3 problems of raster-based methods: LoS are rasterized, causing error; obstructions are evaluated only at raster cell centers; and vegetation is modeled as either fully opaque or fully transparent. They then introduce a method based on TINs and a technique called the visual permeability method. The proposed method starts with a visibility value of 1 (totally visible) and decreases it each time an obstacle is encountered based on its visual permeability and the length of the LoS through the obstacle.

Llobera (2007) builds on Dean (1997) and proposes Equation 1 to develop a more accurate cumulative visibility for LoS. The

raster-based method models the exponential decay in visibility based on Beer-Lambert's attenuation law.

$$p(x) = e^{-k(x)\cdot x} \quad (k \geq 0, x > 0) \tag{1}$$

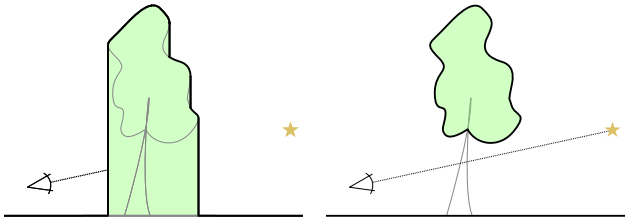
$p(x)$ = the probability of a point being visible,
x = the distance from the observer along the LoS,
$k(x)$ = a function based on the density of vegetation along the LoS.

2.3 Dimensionality

Dimensionality is another limitation of most current methods, which fail to include a fully 3D LoS calculation, relying on simplified representations instead. An overwhelming majority of methods use a 2.5D representation to model visibility (Bartie et al., 2011; Govindaraju et al., 2014; Guth, 2009; Llobera, 2007; Maaiveld et al., 2024; Mistick et al., 2023; Murgoitio et al., 2013).

Depending on the method, this is done through a combination of a Digital Elevation Model (DEM), Digital Surface Model (DSM), Digital Terrain Model (DTM), and sometimes a Canopy Height Model (CHM). These are commonly represented as square, rectangular, or sometimes hexagonal rasters (Maaiveld et al., 2024). Some methods, like Orlof et al. (2024), use Triangulated Irregular Networks (TINs) instead to represent the ground surface. The core limitation of 2.5D models is that for each point on the XY-plane, they can only represent exactly one Z-value.

Figure 3 shows one type of error that is caused by this 2.5D simplification. Pyysalo et al. (2009) show that 21.2% of voxels in their test site contain vegetation and buildings. When represented in 2.5D, all of these voxels are considered full, whereas a 3D representation results in 16 % of the voxels being full and 84 % empty. Trees are particularly susceptible to errors in 2.5D representations since they tend to have many overhangs compared to, for example, terrain and buildings.



(a) (b)

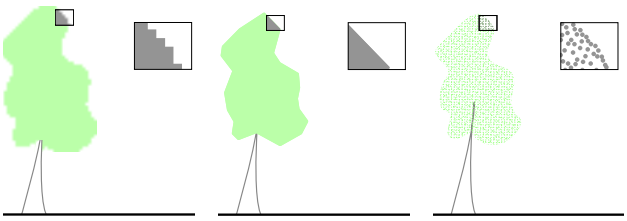
Figure 3: Diagram showing the difference between a 2.5D **(a)** and fully 3D **(b)** representation.

Zhao et al. (2024) proposes a framework to categorize the dimensionality of different methods. They conclude "True 3D requires multiple viewpoints and the fusion of multiple depth maps, which can only be processed using point clouds."

2.4 Representation

Representation of airborne laser scanning (ALS) data is done in a variety of different ways. Generally, the raw input data is first transformed into some intermediate representation like rasters, voxels, or meshes before calculations are done. These intermediate representations trade additional processing time and complexity for a simpler workflow.

While common intermediate representations are widely supported in software, they inevitably introduce a compression error. This means any analysis that follows cannot take full advantage of the high-accuracy scans. Figure 4 shows how direct point cloud representation retains all information compared to derived voxel and mesh representations. Additionally, direct point cloud models offer specific benefits for representing vegetation, which is notoriously hard to represent as meshes or voxels without losing essential information.



(a) (b) (c)

Figure 4: Diagram showing the difference between a voxel model **(a)**, a mesh model **(b)**, and a point cloud model **(c)** of a tree.

Nebiker et al. (2010) and Poux et al. (2016) both argue for "smart" or "rich" point clouds as a new paradigm that enriches point clouds with semantics, structure, and color. Direct point cloud processing workflows preserve high accuracy, and since point cloud acquisition techniques are now widely used, allow for fast production and updating in a fully automatic manner.

Guth (2012) implements an early version of a probabilistic intervisibility algorithm taking into account vegetation directly on point clouds, building on the work of Guth (2009). They show that point clouds offer superior accuracy when compared to raster-based approaches. Zhang et al. (2021) show that taking into account opaque vegetation in these types of visibility calculations directly on point clouds is possible.

3. Methodology

On a high level, the proposed method is made up of the five stages in Figure 5. The *query* stage efficiently retrieves the relevant point cloud data from the source dataset. The *enhance* stage adds additional information and corrections to the data by segmenting vegetation, adding additional points on sparse building facades, and compensating for seasonal differences. The *calculate* stage computes the visibility score for a single LoS between a source and target point. The *produce* stage orchestrates and aggregates these individual LoS rays to construct 2D, 3D, and cumulative viewsheds from multiple observers. Finally, the *apply* stage uses these spatial products for three practical downstream applications: mapping flight height, visibility-aware routing, and optimal lookout site selection.

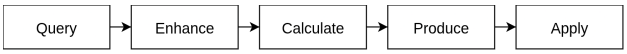


Figure 5: Diagram showing the five main stages of the proposed method

3.1 Query

To handle large-scale point cloud data, the method relies on the Cloud Optimized Point Cloud (COPC) format. COPC builds on the de facto standard LAZ format that structures data into a clustered octree. Unlike traditional point cloud files that require downloading either tiles or the full dataset before processing, COPC allows for efficient range queries using spatial indexing. We use the PDAL (PDAL Contributors, 2025) library to efficiently stream only the specific nodes of the octree that intersect with the user's area of interest. This significantly reduces latency and memory overhead during the querying stage.

The method is designed to work with standard airborne LiDAR. For testing and validation, this research primarily utilizes

the Dutch AHN dataset, alongside the French LidarHD and Canadian CanElevation datasets to ensure generalizability. The AHN is available for download from the TU Delft geotiles project (Natijne, n.d.) and "het Waterschapshuis basisdata" project (het Waterschapshuis, n.d.). Of those, the basisdata portal is preferred because it includes the latest AHN6 data and provides the data as Cloud Optimized Point Cloud (COPC) files.

However, AHN6 does not cover the whole of the Netherlands yet, as it is currently still being collected. Additionally, AHN6 does not include a near-infrared (NIR) color channel. Conversely, the geotiles project (Natijne, n.d.) provided a NIR channel by colorizing versions 1 through 5 of the AHN. This color information is obtained through a post-processing spatial join with aerial imagery. Because the LiDAR point clouds and the aerial photographs are often captured at different times, this process comes with the cost of introducing some temporal incoherence.

3.2 Enhance

3.2.1 Segment vegetation

Since the AHN does not natively have a vegetation class, the "other" class is first segmented into vegetation and non-vegetation. Our method uses the Myria3D deep learning model for this task (Gaydon, 2022). The model is trained and maintained by the French Geographic Institute IGNF and was forked to be used with the AHN dataset.

Myria3D is designed specifically for the semantic segmentation of large-scale aerial LiDAR point clouds. The most recent versions use a neural network architecture called RandLA-Net (Hu et al., 2019). The model is trained on the annotated FRACTAL dataset (Gaydon et al., 2024) on which it achieves an overall IoU of 93.8%.

To classify a scene, the model combines multiple features of ALS data like coordinates, intensity, return number, and the total number of returns. It also optionally uses RGB and NIR color channels to improve accuracy. The model computes a soft classification, outputting a continuous probability distribution across the available classes for every individual point. Our method then uses these probabilities and a threshold to add the vegetation class to the AHN dataset.

3.2.2 Synthesize facades

Given the scanning angle, ALS is notoriously bad at capturing vertical surfaces like facades. To correct for this, we introduce a preprocessing step that adds synthetic points.

The enhancement process, detailed in Algorithm 1, relies on the assumption that most roofs are supported by vertical facades extending from their perimeter to the ground below. The algorithm begins by isolating roof points using class labels and normals. Next, it clusters them and computes the 2D boundary footprint of each roof. To bring this footprint back into 3D space, the Z-coordinates for the boundary vertices are retrieved from their nearest neighboring points in the original roof cluster. Finally, the algorithm samples points along these 3D boundary edges at a specified density, generating vertical columns of points from the roof elevation down to the ground underneath.

Input: Point cloud P, normal threshold t, sampling density d
Output: Enhanced point cloud with synthetic facades
$P_{\text{building}} \leftarrow$ filter P for points classified as buildings
Compute surface normals for P_{building}
$P_{\text{roof}} \leftarrow$ filter P_{building} where NormalZ $> t$
$C \leftarrow$ cluster P_{roof} using DBSCAN
$P_{\text{facade}} \leftarrow \emptyset$
$K_{\text{ground}} \leftarrow$ build KDTree of ground points in P
for each cluster c in C **do**
    $B \leftarrow$ compute 2D concave hull boundary of c
    for each 2D vertex v in B **do**
        $v.z \leftarrow$ Z-coordinate of the nearest 3D point in c
    for each consecutive pair of 3D vertices (v_1, v_2) in B **do**
        $E \leftarrow$ sample points along the edge (v_1, v_2) at density d
        for each sampled point p in E **do**
            $z_{\text{roof}} \leftarrow$ interpolate roof elevation between v_1 and v_2 at p
            $z_{\text{ground}} \leftarrow$ query nearest ground elevation from K_{ground} at $(p.x, p.y)$
            if $z_{\text{roof}} > z_{\text{ground}}$ **then**
                $P_{\text{col}} \leftarrow$ generate points from z_{ground} to z_{roof} at density d
                Append P_{col} to P_{facade}
return $P \cup P_{\text{facade}}$

Algorithm 1: EnhanceFacades (P, t, d)

3.3 Calculate

Before calculating the attenuation for a Line of Sight (LoS), we must first load the relevant points into the computer's memory. Given the discrete floating-point coordinates of LiDAR data, the theoretical LoS will not exactly intersect any of the measured points. Therefore, we add a margin to our query, treating points within this margin as intersections. One can imagine this query volume as a cylinder with some radius r, where the centerline is the theoretical LoS.

An alternative frustum-shaped model that could offer benefits when modelling human vision is explored in the discussion. However, it was not used to generate the presented results due to the focus and scope of this research. Regardless of LoS shape, the core algorithm walks along the LoS in discrete steps. At each step, it uses optimized ball queries to calculate visibility between a source point and a target point.

3.3.1 Spatial index

To efficiently select the points within this cylinder, we organize the point cloud into a k-d tree (Bentley, 1975) and use highly optimized spherical queries. While a comprehensive comparison of spatial indexes (Samet, 1984) is not the core objective of this research, k-d trees offer theoretical benefits over an octree structure (Meagher, 1982) for this specific use case.

First, urban point clouds exhibit highly non-uniform point density. Buildings and terrain are densely sampled, while the surrounding airspace is largely empty. Octrees evenly partition and hierarchically refine space, causing a tree with inconsistent depth in this case. In contrast, the selected k-d tree implementation from the Scipy library (Virtanen et al., 2020) evenly partitions data and lets us guarantee a balanced tree upon construction.

Second, the k-d tree is fundamentally optimized for spherical queries. When querying a thin cylinder through an octree, the geometry pierces hundreds of rigid bounding boxes but rarely engulfs them, forcing the algorithm into deep leaf-node traversals.

Third, while octrees are highly efficient for dynamic datasets that require constant spatial updates, our analysis is performed on static LiDAR data. Therefore, the algorithmic priority shifts completely to query speed rather than insertion time. In this context, optimized k-d trees are among the best-performing structures for spatial queries in 3D point clouds (Elseberg et al., 2012).

3.3.2 Optimal query radius

For optimal computational speed, one must consider two trade-offs. First, the trade-off between performing many small queries that tightly fit the cylinder versus fewer large queries that include more volume outside the cylinder.

Second, the cost of performing many small queries that introduce more overlap between spheres versus fewer large queries that reduce overlap. While significant mathematically, this effect is negligible compared to the first in our implementation. Filtering irrelevant points outside the cylinder uses distance calculations that are far more expensive than point deduplication for any reasonable step size.

In Figure 6, R is the query sphere radius, r the cylinder radius, s the step size between two adjacent sphere centers, and x the radius of the intersection of two adjacent query spheres. Additionally, it marks the overlapping areas in the 2D diagram in orange and the queried area outside the cylinder in red.

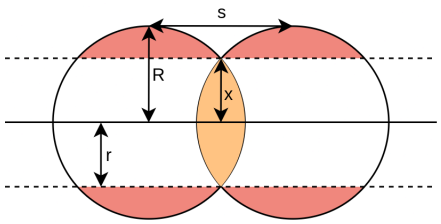


Figure 6: 2D diagram of the cylindrical volumetric line of sight model with named variables.

To guarantee full coverage while minimizing query volume outside the cylinder, the intersection radius x must equal the cylinder radius r. By substituting r into the Pythagorean theorem, the optimal query radius R is given in Equation 2.

$$R = \sqrt{\frac{s^2}{4} + r^2} \tag{2}$$

3.3.3 Early exit

While optimizing the query distance s minimizes geometric waste for an unobstructed LoS, realistic queries rarely travel their full distance before intersecting an obstacle.

To take advantage of this, we introduce chunking to the algorithm. The algorithm walks the LoS in discrete steps, retrieving the relevant points for each. If a step exceeds the threshold for buildings or terrain, the algorithm triggers an early exit. Every remaining step in the LoS is marked as fully invisible, which takes a fraction of the computation time without the need to query the input point cloud.

This mechanism ensures that the algorithm scales highly efficiently in dense environments, as the runtime becomes proportional to the distance to the first obstacle rather than the total theoretical length of the LoS.

3.3.4 The algorithm

Using these optimizations, we evaluate the attenuation for each ball query as described in Algorithm 2 and shown in Figure 7. The algorithm walks the LoS in discrete steps and checks if the local density of any point class exceeds its respective threshold (T). These density thresholds exist to filter out minor classification errors, sensor noise, or sparse, transient objects like birds. Furthermore, a volumetric cylinder grazing the far edge of a building or capturing a single sparse branch should not fully occlude the view. By requiring the point density to exceed a predefined limit, the algorithm prevents these anomalies from causing false occlusions.

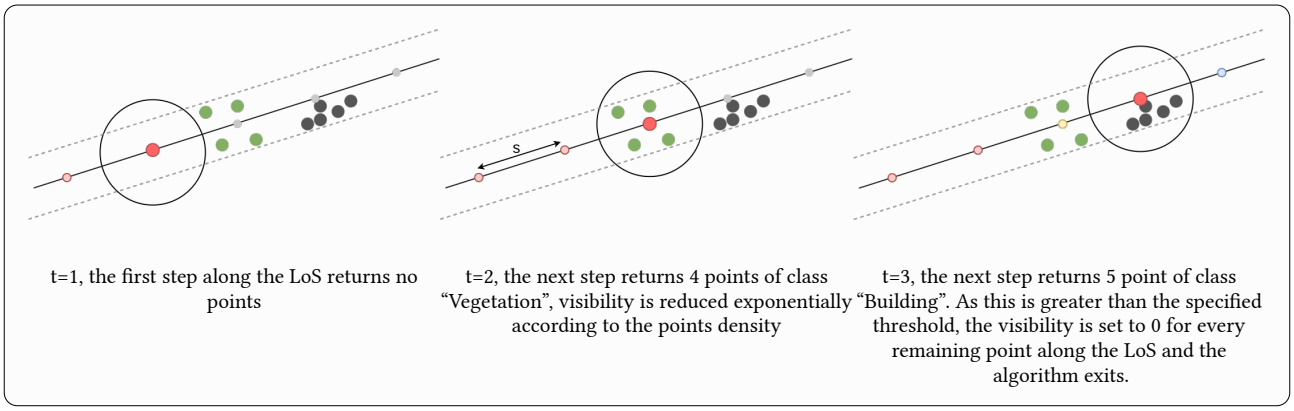
When a threshold is exceeded, the algorithm reacts based on the classification of the points. If the points are classified as solid obstacles (either building or ground), visibility is immediately set to zero, and the algorithm triggers its early exit. If the points are classified as vegetation, visibility is reduced exponentially according to Equation 1. While ALS provides exact coordinates, typical resolutions only sample a fraction of the complex canopy. Because the space between LiDAR points is not guaranteed to be empty, we use the captured point density (ρ_{veg}) as a proxy for physical density to calculate this attenuation.

Input: LoS cylinder C, point cloud P, KDTree K, thresholds T_{terrain}, T_{bldg}, T_{veg}, step size s, and attenuation coefficient k
Output: v: final visibility score, V: array of step visibilities
$v \leftarrow 1.0$
for each step s in C **do**
  $P_{\text{sphere}} \leftarrow$ query a sphere in K for points in the current step
  $P_{\text{step}} \leftarrow$ filter P_{sphere} to points strictly inside C
  if P_{cyl} is empty **then**
    continue
  $\rho_{\text{terrain}} \leftarrow$ # terrain points in P_{step}
  $\rho_{\text{building}} \leftarrow$ # building points in P_{step}
  $\rho_{\text{veg}} \leftarrow$ # vegetation points in P_{step}
  if $\rho_{\text{terrain}} \geq T_{\text{terrain}}$ **or** $\rho_{\text{building}} \geq T_{\text{building}}$ **then**
    //– Line of sight is completely blocked by solid obstacles
    return 0.0, V
  if $\rho_{\text{veg}} \geq T_{\text{veg}}$ **then**
    //– Attenuate visibility through vegetation via Beer-Lambert law
    $v \leftarrow v \cdot \exp(-k \cdot \rho_{\text{veg}} \cdot s)$
  $V[s] \leftarrow v$
return v, V

Algorithm 2: Visibility (C, P, K)

3.3.5 Seasonal differences

A challenge that comes with utilizing large-scale ALS datasets is the timing of the data capture. Datasets collected in winter will underestimate the visual obstruction of vegetation in summer and vice versa.



t=1, the first step along the LoS returns no points

t=2, the next step returns 4 points of class "Vegetation", visibility is reduced exponentially according to the points density

t=3, the next step returns 5 point of class "Building". As this is greater than the specified threshold, the visibility is set to 0 for every remaining point along the LoS and the algorithm exits.

Figure 7: Method diagram

In our adaptation of Beer-Lambert's law, we can partially compensate for this effect using the attenuation coefficient k. In our model, k is made up of the local density of vegetation points (ρ) and a base attenuation constant (c), such that $k = c \cdot \rho$.

Equation 3 shows that since attenuation exponent is a linear product, mathematically scaling the base attenuation constant c by a seasonal scaling factor α is the same as scaling the physical point density ρ by α.

$$\begin{aligned} p(x) &= e^{-c \cdot \rho \cdot x} \\ p_{\text{physical}(x)} &= e^{-c \cdot (\alpha \cdot \rho) \cdot x} \\ p_{\text{math}(x)} &= e^{-(\alpha \cdot c) \cdot \rho \cdot x} \\ p_{\text{physical}(x)} &= p_{\text{math}(x)} \end{aligned} \tag{3}$$

This mechanism gives us a convenient way to simulate seasonal variations by adjusting the attenuation constant c. This purely mathematical approach also avoids increasing file sizes, prevents the degradation of query performance, and introduces no artificial geometric errors into the data.

3.4 Produce

The proposed methodology for a single LoS calculation is the foundation for multiple types of visibility analyses. By repeating the core calculation, the method can produce high-level spatial products like 2D, 3D, and cumulative viewsheds from multiple observers.

2D viewsheds are generated by repeating LoS calculations between an observer and a series of target points. There are two primary approaches for defining these targets. The first is boundary sampling, where target points are distributed solely along the perimeter of the AOI. Because our algorithm evaluates visibility continuously along the ray, casting rays to the perimeter effectively fills in the enclosed area. This method is highly efficient and works well for relatively flat environments.

However, the second method, full grid sampling, computes an individual LoS to every cell in a 2D grid covering the AOI. While this is computationally slower, it is more rigorous and prevents errors in highly varied terrain where intermediate elevation changes might obscure local depressions. Unlike traditional binary viewsheds, both sampling methods result in a probabilistic viewshed that creates a continuous gradient of visibility across the environment.

3D viewsheds expand the analysis into the vertical dimension. It is important to note that this does not require a computationally expensive evaluation of every individual point within a 3D grid. Instead, the method operates efficiently by calculating LoS rays from the source point to a bounding surface of target points that encloses the AOI.

While a rectangular bounding box of target points can be used, which conveniently ensures that all horizontal cross-sections of the resulting data have the same area, it can introduce "stretching" artifacts due to the disproportionately long rays reaching into the corners. To prevent this, the target points can be distributed over a bounding hemisphere, ensuring uniform ray lengths and consistent sampling density. Because visibility scores are recorded at discrete steps along the length of every ray as it travels toward this boundary, the result is a complete, dense 3D "visibility space" that maps which parts of the volume are exposed to the observer.

Cumulative viewsheds from multiple observers can be constructed by aggregating several individual viewsheds together. For this aggregation, we can use several methods like taking the maximum, minimum, mean, or sum of the individual visibilities. Each of these may be most suitable for a particular type of mission. For example, when being spotted by any single observer immediately compromises the mission, aggregating using the maximum visibility score is a logical choice. But when setting up air defense, it may be optimal to use minimal aggregation to ensure full coverage.

3.4.1 Visualize

We have implemented three distinct ways to visualize the viewshed in 3D space: direct point cloud export, voxelization, and web-based dissemination.

First, users can save the calculated viewsheds directly as COPC files using PDAL. This allows users to load the results directly into standard point cloud processing software, such as CloudCompare or ArcGIS. Because the visibility score for each step along the LoS is saved as a custom attribute, visibility can be easily visualized and filtered using custom color ramps and thresholds.

Second, users can export voxelized representations of the viewshed. Visibility scores are aggregated by either maximum of minimum, depending on the scenario, making the data more structured. This volumetric approach is particularly useful to speed up downstream tasks that scale with point density, such as 3D pathfinding algorithms for drones or helicopters.

Finally, to make the 3D results accessible to stakeholders who do not have access to specialized GIS software, the outputs can be visualized through a web-based interface. Using the open-source

Potree renderer together with optimized COPC files, we provide a 3D viewer directly in the browser[5].

While these 3D visualizations are suitable for many applications, they cause a breaking issue for the application of airborne mission planning: a 3D representation style does not easily integrate into current mission planning processes. Mission planning involves combining a number of different factors that are used by experts to make informed decisions. While future versions of these systems might move to a fully 3D representation, they are not in widespread use today. Therefore, the method is also flexible enough to export calculated analyses as GeoTIFF raster files. By taking sections of the volumetric viewshed, raster outputs can be presented to users as so-called map overlays. Here, information is presented in 2D by mapping it over the ground surface. Crucially, this approach allows for integration of the visibility analysis with other effects that need to be considered.

3.5 Apply

To show the kind of high-level analysis that is possible using our method, we present three distinct prototypes that could be useful in the context of military airborne mission planning. These are: safe flight height maps, visibility-aware pathfinding, and optimal lookout site selection. Because our method calculates visibility on a continuous scale rather than a traditional binary value, selection of specific risk thresholds is required for producing them. These thresholds allow mission planners to explicitly define the maximum acceptable probability of detection based on the risk profile of a given operational scenario.

3.5.1 Flight height map

The flight height map shows the maximum altitude an aircraft can maintain while remaining hidden. As described in Algorithm 3, the user defines a maximum acceptable risk threshold between zero and one. The algorithm then initializes a 2D grid over the AOI and computes the height where that value is exceeded for each cell.

Input: 3D viewshed points P, Area of Interest A, risk threshold t, resolution r, ceiling $Z_{\max}$
Output: M: a 2D raster mapping safe flight heights (HAG)
Initialize a 2D grid G over A with cell size r
for each cell c in G **do**
  if c is outside A **then**
    continue
  $z_{\text{safe}} \leftarrow Z_{\max}$
  $P_c \leftarrow$ filter P to points within the 2D boundaries of c
  Sort P_c by Z-coordinate in ascending order
  for each point p in P_c **do**
    if $p_{\text{visibility}} \geq t$ **then**
      //– Maximum risk threshold exceeded; record height and exit
      $z_{\text{safe}} \leftarrow p_z$
      break
  $c_{\text{ground}} \leftarrow$ sampled ground elevation at c
  $M[c] \leftarrow z_{\text{safe}} - c_{\text{ground}}$ //– Record Height Above Ground (HAG)
Save M to disk as a GeoTIFF
return M

Algorithm 3: SafeFlightHeight (P, A, t, r, $Z_{\max}$)

3.5.2 Visibility-aware routing

Visibility-aware routing uses the volumetric viewshed to assess different optimal paths through open 3D space. To achieve this, we construct a 3D navigational graph from the point cloud, explicitly marking voxels occupied by solid structures or terrain as blocked.

The routing is implemented using a modified A* algorithm to traverse this graph. The edge cost is defined as a weighted sum of the length and the visibility score along that edge, multiplied by a user-defined constant.

Additionally, we introduce a custom heuristic that applies a distance-dependent penalty to the visibility score. The reasoning is that being detected early in an approach is significantly more dangerous than being detected late, as it gives the target more time to react.

By tuning these cost weights, mission planners can plan routes that prioritize either the shortest travel time, the minimum overall exposure, or the minimum reaction time for the adversary. The algorithm presents the user with a selection of optimal routes. These routes are sampled by sweeping the relative weight of the visibility parameters.

3.5.3 Optimal lookout site selection

While the previous applications focus on visibility from the perspective of an infiltrator, the methodology is also applicable for optimizing observation from a defender's perspective. Algorithm 4 identifies the point where a lookout post or sensor has maximum visibility of a specific area like a road. It does this using a relatively simple search where LoS are computed between the target road to observe and each of the sampled points in the AOI.

[5] https://gist.bk.tudelft.nl/apps/PointCloudLineOfSight/

Input: 3D point cloud P, Area of Interest A, trail path T, resolution r, threshold t
Output: p_{best}: optimal observer location
$T_{\text{targets}} \leftarrow$ sample points along trail T
Initialize a 2D grid G over A with cell size r
$c_{\text{best}} \leftarrow -1.0$
$p_{\text{best}} \leftarrow$ null
for each candidate c in G **do**
  //– Compute how many targets are visible
  $d_{\text{count}} \leftarrow 0$
  for each target d in D **do**
    $v_{\text{score}} \leftarrow$ calculate intervisibility from c to t using P
    if $v_{\text{score}} > t$ **then**
      $d_{\text{count}} \leftarrow d_{\text{count}} + 1$
  //– Update best lookout
  $c_{\text{score}} \leftarrow \left(\frac{v_{\text{count}}}{T_{\text{targets}}}\right) \times 100$
  if $c_{\text{score}} > c_{\text{best_score}}$ **then**
    $c_{\text{best_score}} \leftarrow c_{\text{score}}$
    $c_{\text{best}} \leftarrow c$
return c_{best}

Algorithm 4: OptimalLookout (P, A, T, r_c, r_f, h_o, t)

To ensure an acceptable runtime, this optimization can be performed using a multi-pass, coarse-to-fine search. An initial low-resolution pass identifies the most promising general area. The algorithm then performs a second search pass around the best known points to further refine the optimal placement.

4. Results

4.1 Products

To generate the 2D viewshed, target points were sampled along the boundary of the AOI at exactly 1.8 meters Height Above Ground (HAG). This specific height offset is used to simulate the real-world perspective of an observer at eye level.

Figure 8 shows an example viewshed mapped directly onto the point cloud. The output utilizes a continuous color scale: red indicates areas of full visibility ($p = 1$), blue indicates areas of total occlusion ($p = 0$), and the intermediate orange hues represent areas of partial occlusion caused by semi-transparent vegetation.

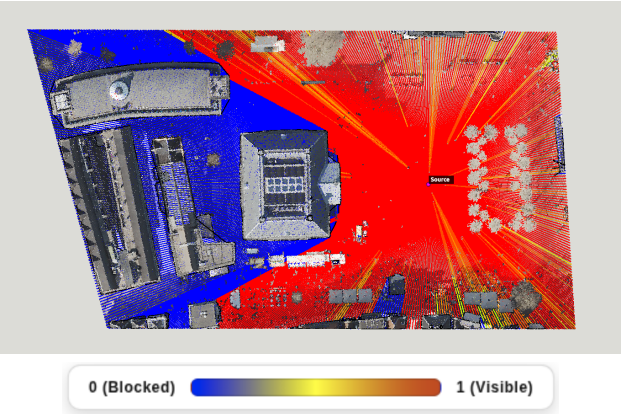


Figure 8: 2D probabilistic viewshed mapped over the point cloud. Red indicates full visibility, blue indicates total occlusion, and orange gradients represent partial occlusion through vegetation.

Figure 9 shows how the viewshed methodology can be extended into three dimensions. To generate this volumetric representation, target points are sampled over a 3D surface grid that bounds the area of interest. The observer's position is once again set at 1.8 meters HAG. By evaluating the continuous visibility along the rays cast from the observer to this outer surface, the algorithm effectively maps the entire internal 3D visibility space.

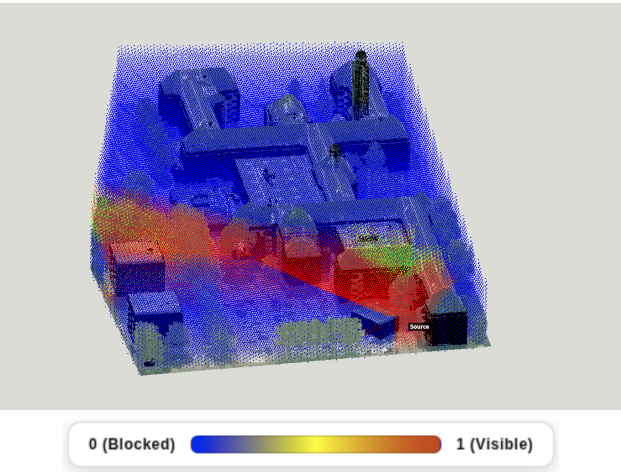


Figure 9: A volumetric viewshed visualizing the full 3D visibility space around an observer.

Figure 10 displays a cumulative volumetric viewshed. In this visualization, multiple observer source points are selected within the volume, and their combined overlapping coverage is computed.

This specific representation uses maximum aggregation, providing a worst-case exposure map. These different products can then serve as a base for a variety of downstream automated analyses as discussed in the next section.

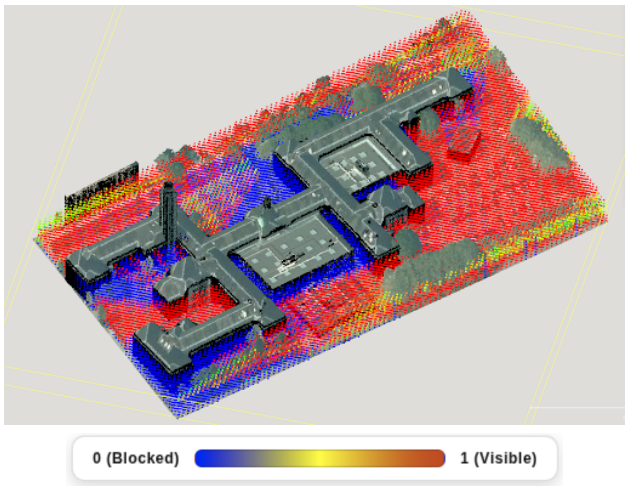


Figure 10: A cumulative volumetric viewshed representing the maximum aggregated visibility from multiple observer locations, providing a worst-case exposure map.

4.2 Applications

4.2.1 Flight height map

The algorithm outputs both a point cloud (Figure 11) and a 2D raster map (Figure 12) representing the safe flight ceiling as a height above ground (HAG) z-value. The main benefits are the ability for users to set an appropriate risk threshold and removing the limitation of previous relatively coarse raster resolutions.

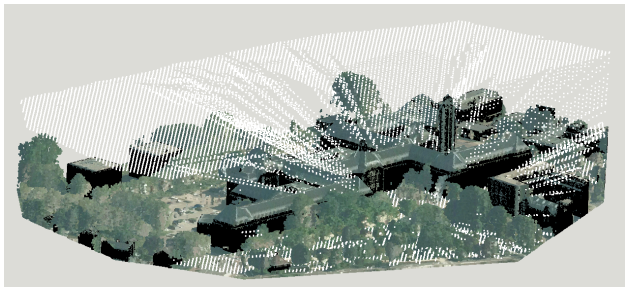


Figure 11: Flight height result exported as COPC visualized in CloudCompare.

For operational use, the 2D GeoTIFF provides an intuitive map overlay. As seen in Figure 12, the observer location is indicated by the white dot. The continuous color ramp represents the safe flight ceiling, transitioning from red (low maximum flight height, meaning high risk of detection even near the ground) to blue (high maximum flight height, indicating a safe approach vector).

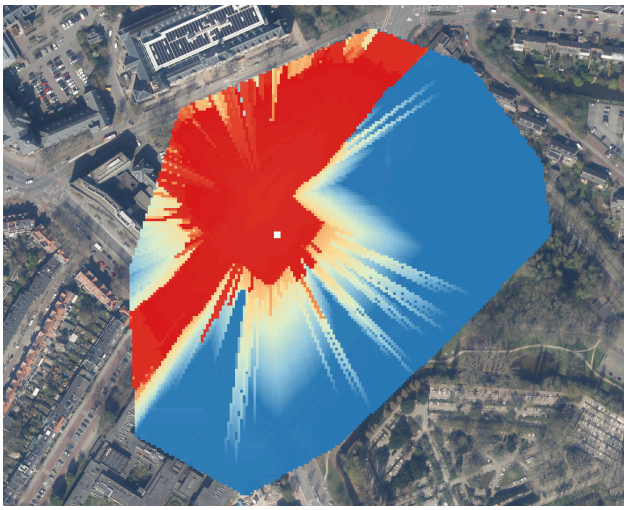


Figure 12: Flight height result exported as GeoTIFF visualized in QGIS

4.2.2 Visibility-aware routing

By varying the cost weights between shortest physical distance and minimal visibility exposure, the routing algorithm generates a set of path options. As shown in Figure 13, the observer (target) is located at the far end of the route.

The resulting visualization presents the user with two distinct routes. Each one is computed with a different balance between prioritizing the shortest path and minimal visibility. Offering multiple choices allows mission planners to select the optimal trade-off between stealth and travel time for their specific scenario.

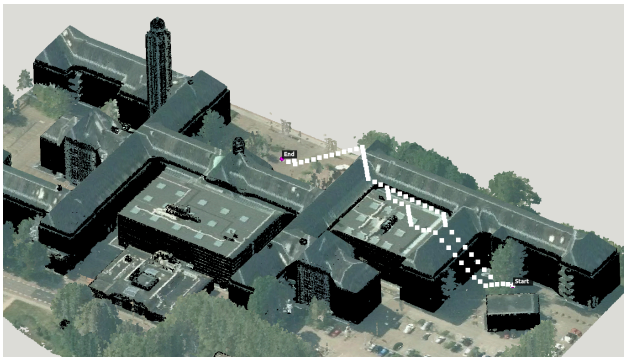


Figure 13: Visibility-aware routing result.

4.2.3 Optimal lookout site selection

Applying the methodology described in Algorithm 4, we can map the effectiveness of candidate observation posts over a designated target area. In this scenario, the objective is to maximize the visual coverage of a specific road.

Figure 14 shows the result of a two-pass coarse-to-fine search. The resulting point cloud is colored according to the lookout effectiveness score, with the color ramp ranging from blue (indicating low visibility of the road) to red (indicating high visibility of the path). Using this method, the algorithm successfully identified an optimal observer placement on a small hill overlooking the route.

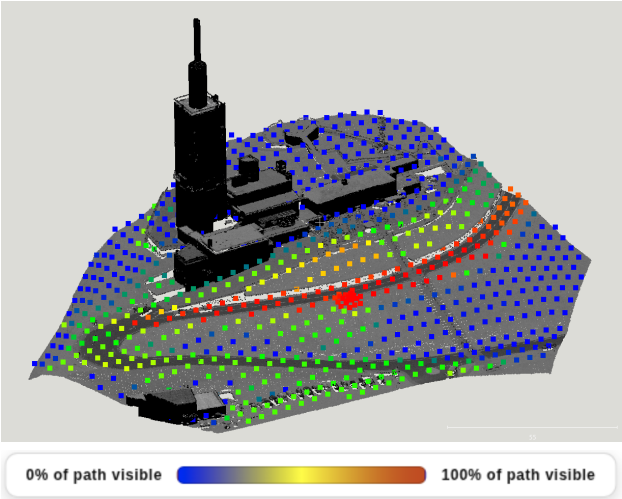


Figure 14: 3D visualization showing the evaluated visibility scores for candidate lookout locations across the area of interest.

4.3 Evaluation

4.3.1 Raster method

To evaluate our method, we use a case study to compare it with industry-standard 2.5D raster methods using standard GIS software (QGIS). While raster-based tools are currently the most widely adopted method for operational mission planning, their reliance on Digital Surface Models (DSMs) introduces significant limitations.

As shown in Figure 15, there are significant differences in the resulting viewsheds between the two methods, even when operating on the same underlying LiDAR data and final resolution. Besides being limited to 2.5D and the 0.5m resolution of the DTM, we can also see that the raster-based method treats trees as completely opaque 2.5D obstacles. This results in a severe underestimation of the viewshed.

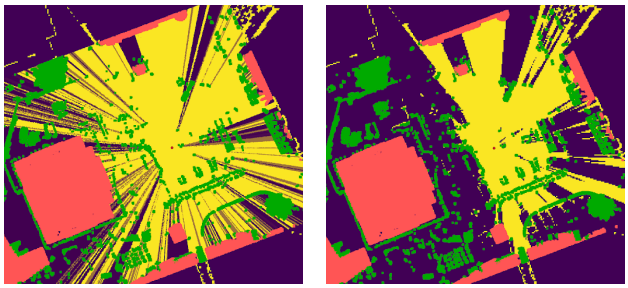


Figure 15: Comparison of our method (left) to the raster-based method using QGIS (right)

Figure 16 illustrates the difference in accuracy between the two methods. The raster-based method converts the point cloud into a standard 2.5D raster before calculating the viewshed, resulting in information loss. Conversely, our method calculates the viewshed in 3D space first, and only rasterizes the final visibility scores. This "calculate first, rasterize later" approach retains significantly more information and accuracy than current industry-standard algorithms.

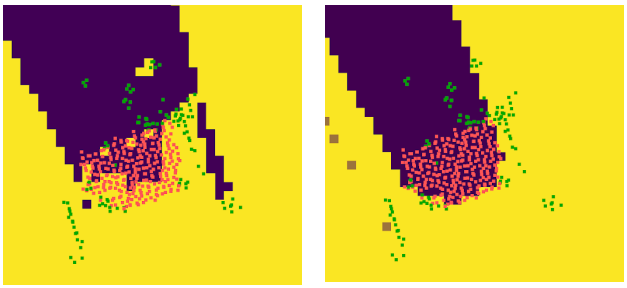


Figure 16: Detailed comparison of viewshed methods in QGIS (left) and our method (right)

4.3.2 3DBAG benchmark

To evaluate the accuracy of our method in 3D, we construct a synthetic point cloud from the 3DBAG. The 3DBAG is itself based on the AHN but uses governmental property maps to convert it into highly detailed 3D building meshes. We sample points on these meshes to create a benchmark that is unrealistically dense, perfectly uniform, and does not suffer from occlusion.

Because the viewsheds for this validation are only influenced by buildings, we treat visibility as a binary classification task. A target is either visible (positive class) or occluded (negative class). We then compare our method to the benchmark in terms of precision (Figure 17), recall (Figure 18), and F1-score (Figure 19) to evaluate the effect of different LoS radii and facade enhancement densities. These metrics are defined as follows:

- **Precision:** Out of all the target points our algorithm claimed were visible, how many were actually visible in the 3DBAG benchmark? A drop in precision means the algorithm is missing solid objects.
- **Recall:** Out of all the target points that are actually visible in the 3DBAG benchmark, what percentage did our algorithm successfully find? A drop in recall means the algorithm is falsely blocking clear lines of sight.
- **F1-score:** The harmonic mean of precision and recall, providing a single metric to evaluate the overall balance of the algorithm.

Figure 17 shows the impact of these parameters on the algorithm's precision. When calculating visibility on raw, unenhanced point clouds using a small LoS radius, precision drops significantly. This is caused by an effect we call "needle-threading," where the thin LoS cylinder passes through the gaps between the LiDAR points on vertical facades. Increasing the LoS radius resolves this by thickening the ray until it is guaranteed to hit a point, resulting in perfect precision. The graph also demonstrates that denser facades allow the algorithm to achieve perfect precision at smaller radii compared to sparse facades.

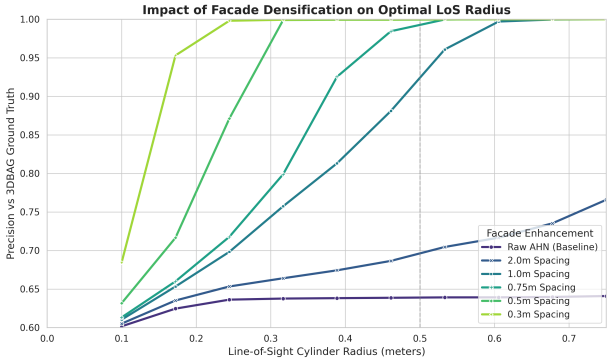


Figure 17: Impact of facade densification and LoS radius on precision.

However, while increasing the LoS radius increases precision, it degrades recall, as shown in Figure 18. Large cylinders are more likely to hit the sides of buildings, overhanging roofs, or nearby obstacles that do not actually obstruct the true geometric line of sight. The graph also shows that facade densification comes at the cost of slightly lower recall. This is because the facade enhancement step will, like any processing, always introduce some error.

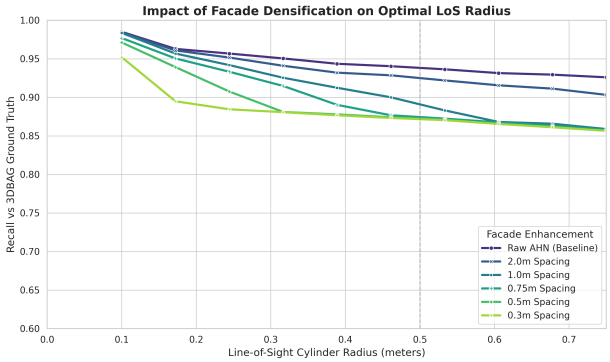


Figure 18: Impact of facade densification and LoS radius on recall.

Figure 19 combines these competing metrics by calculating the F1-score. On the left side of the curve, rays are small, and the loss of precision becomes dominant. On the right side of the curve, rays are large, and the loss of recall becomes dominant. Introducing facade densification shifts the F1 peak significantly towards smaller radii and higher scores. By enhancing building facades, we trade a small amount of recall for a large improvement in precision.

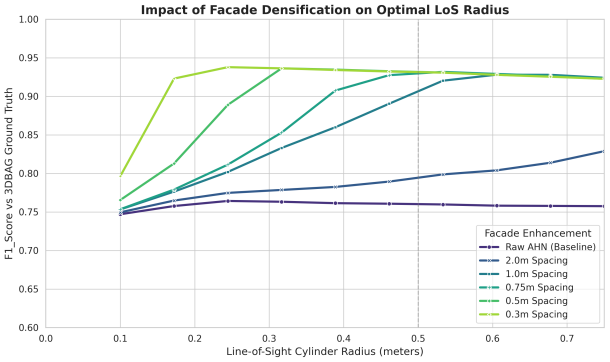


Figure 19: Impact of facade densification and LoS radius on F1-score.

Based on this evaluation, we utilize a LoS cylinder radius of 15 cm for all subsequent calculations. This parameter reflects the physical limitations of the underlying AHN dataset. First, the AHN possesses an average point density of roughly 10 pts/m², which translates to a mean point spacing of approximately 30 cm. Second, decreasing the diameter any further approaches the positional accuracy limits of the ALS measurements themselves. Modeling visibility at a spatial resolution finer than the uncertainty of the raw sensor data is mathematically unsound.

4.3.3 Seasonal differences

As established in Section 3.3.5, the visual impact of seasonal changes can be simulated by scaling the attenuation constant, as shown in Equation 3. To validate this mathematical approach, we compare it to an increase in points simulating a higher vegetation density. This physical method artificially simulates summer conditions by duplicating existing leaf-off vegetation points and applying a slight spatial jitter to disperse them, thereby directly increasing the physical point density (ρ).

Figure 20 shows that increasing the attenuation constant c achieves the same effect as physically increasing the density of vegetation by duplicating internal points.

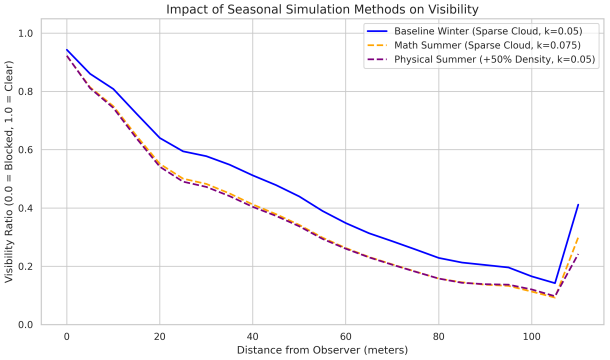


Figure 20: Comparison of baseline leaf-off winter data with mathematical and physical density adjustments to simulate summer conditions.

4.3.4 Vegetation impact

To evaluate the effect of our semi-transparent vegetation model, we analyze the decay of visible airspace as it radiates outward from an observer. The y-axis in the following figures represents the voxel-equivalent cumulative volume. This means that two voxels, both with a visibility score of 0.5, are equivalent to one fully visible voxel.

We compare three distinct vegetation modeling strategies: a transparent model that ignores vegetation entirely, a binary model that treats all vegetation as opaque, and our proposed probabilistic model.

Figure 21 illustrates this comparison in a highly vegetated, urbanized area. As expected, the transparent model overestimates visibility while the binary model severely underestimates it. The probabilistic model provides a middle ground and realistically models how visibility gradually degrades. It also shows how, for source point A (top), the difference between models is much more noticeable compared to source point B (bottom). This is logical as point A is located in a far more densely vegetated part of the AOI.

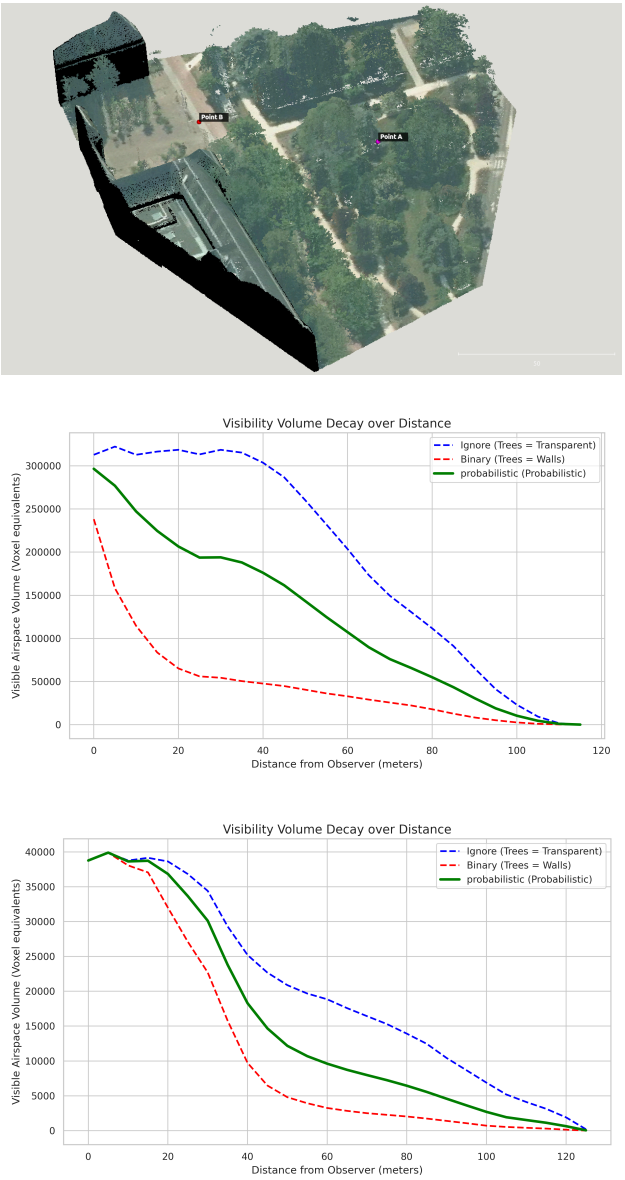


Figure 21: Comparison of visible airspace volume decay using different vegetation models. Two source points A (top) and B (bottom) are shown.

Figure 22 evaluates a much sparser environment, resulting in smaller differences between the three models. However, the tactical significance of the probabilistic model in this environment could actually be higher. As there are fewer obstacles overall, trees contribute an even more significant percentage of the available cover.

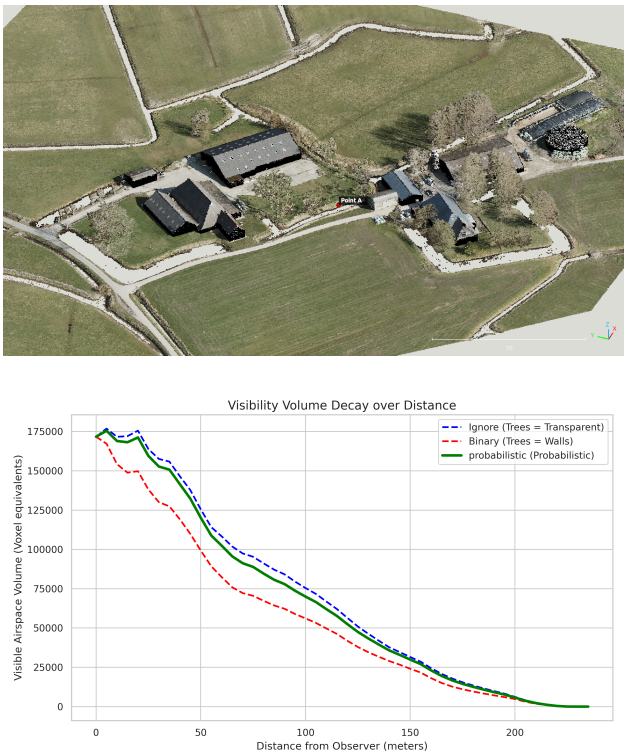


Figure 22: Comparison of visible airspace volume decay using different vegetation models for a high-visibility area of interest.

4.3.5 Performance

To optimize query performance, we empirically determine the best value for the step size s by measuring average runtime over a normalized set of LoS queries, each averaged over 100 runs. Figure 23 shows a typical result for one set of LoS with a 15 cm radius on a log-log scale.

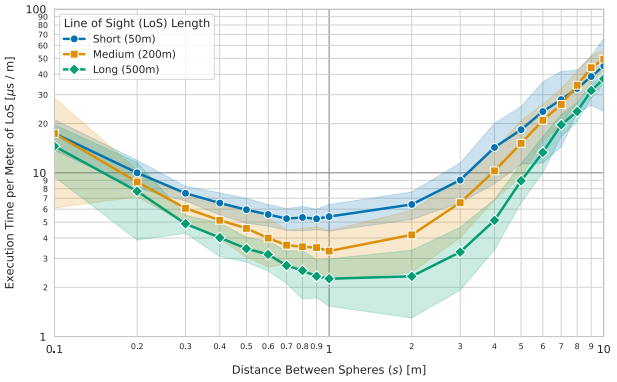


Figure 23: Query performance for different step sizes averaged over 100 runs. Note that both axes are on a logarithmic scale.

While the exact minimum depends on the specific dataset and hardware, our benchmark shows an optimal query distance of approximately one meter for a cylinder with a 15 cm radius. As s approaches zero, the number of queries required approaches infinity and the overlap between them approaches 100%, causing execution time to explode. For larger values of s, the required sphere radius R expands, resulting in cubic growth of the queried volume. This causes increasingly more points outside the cylinder to be included, which bottlenecks the CPU

during the geometric filtering stage. A complete mathematical derivation of these volumetric fractions as a function of the query radius is provided in Section .

Performance is significantly improved compared to previous methods for direct point cloud visibility. Zhang et al. (2021) uses a similar technique where the algorithm walks the LoS in steps and reports a performance of between 2 and 5 LoS calculations per second. Depending on settings, our method processes roughly 1000 LoS per second of runtime on the same area and the same dataset. While we are not able to compensate for some effects like hardware improvements, as they are not described in the paper, our method conservatively achieves a speedup of at least two orders of magnitude.

This increase in performance is mainly due to three strategies. First, rather than calculating visibility independently for every discrete target point, the algorithm evaluates many points along the entire Line of Sight (LoS) in one iteration. This means that for a given 3D AOI, the algorithm only has to calculate the number of LoS proportional to its surface area instead of its volume.

Second, the algorithm walks the LoS in discrete steps and extracts points using sequential spherical queries. Because a sphere is a highly compact and perfectly symmetrical geometric object, it allows spatial index structures to operate efficiently. In our implementation, the local points are organized into a k-d tree, allowing query times with an average time complexity of $O(\log N)$ with respect to the point count N. However, because the algorithm searches with a compact object, this step-based spherical approach means other hierarchical index structures, such as an octree, would also work well.

Third, the moment a ray intersects a solid obstacle that exceeds the structural threshold, the algorithm terminates and assigns a visibility score of zero to all subsequent steps along that ray. Because of this early-exit mechanism, the execution time for a single ray is largely proportional to the distance to the first solid obstacle, rather than the total theoretical length of the ray. The algorithm only processes the full theoretical length of the ray during worst-case scenarios, such as an entirely unobstructed view.

With the core visibility calculation optimized, the necessary preprocessing steps are now a significant part of the runtime, as they scale linearly with the number of points. However, conventional GIS pipelines also require processing time to filter, interpolate, and convert raw ALS data into rasterized Digital Surface Models (DSMs). Still, these traditional pre-processing methods are highly optimized, and while direct benchmarks are unavailable, they likely currently outperform our method in initial data ingestion.

We identify two basic approaches for handling this pre-processing. The first approach is precomputing the entire dataset beforehand and caching it to be used when analysis is needed. This is by far the most common strategy in current DTM/DEM workflows. The main benefit of this approach is that the bulk of the computation can be done in a low-stakes, controlled environment. The drawback is the upfront computational cost of processing data that will likely never be tactically relevant.

The second approach is preprocessing only the relevant parts of the dataset on the fly. While this reduces the overall runtime by exclusively processing the necessary data within the AOI, it shifts that runtime delay to a higher-stakes, operational moment. Additionally, this approach may duplicate calculations if multiple users independently analyze an overlapping AOI.

5. Discussion

5.1 LoS shape

The results generated in this research rely on a cylindrical Line of Sight (LoS) model. This uniform volumetric buffer is a practical, consistent method for capturing discrete LiDAR points along a theoretical mathematical vector. However, while computationally robust, the concept of a LoS is not a perfect model for human vision, introducing several geometric and physical trade-offs when compared to a diverging, frustum-shaped (cone) model.

From a biological perspective, human vision operates much more like a frustum. The retina captures light in receptor cells (rods and cones), and tracing the light entering a single cell back into the 3D environment forms a narrow, expanding cone.

Furthermore, from a spatial indexing perspective, frustums offer significant advantages when computing comprehensive 3D viewsheds. If an observer casts multiple frustum-shaped rays outward, these volumes pack together to cover more of 3D space while skipping fewer points. Casting cylindrical rays from a single point, the geometries overlap near the observer and diverge as they get further away.

Figure 24 illustrates the difference between an individual cylindrical and frustum-shaped LoS.

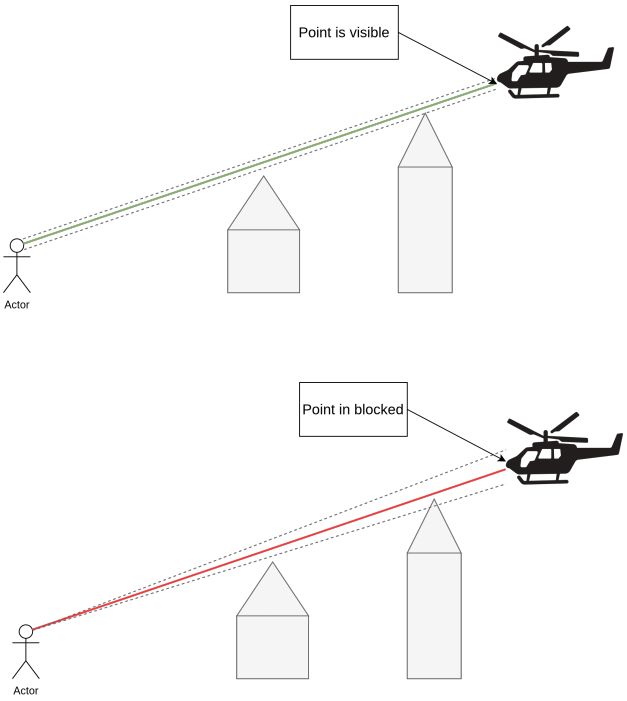


Figure 24: Diagram comparing a cylindrical LoS shape (top) to a diverging frustum LoS shape (bottom)

Together with these biological and spatial packing advantages, implementing a frustum-shaped approximation introduces three issues for our specific density-based attenuation algorithm:

First, an expanding radius inherently breaks the principle of intervisibility symmetry. In theoretical geography, intervisibility is generally assumed to be bi-directional (if Point A can see Point B, Point B can see Point A). In reality,

visibility is often asymmetric. For example, an observer standing immediately behind a tree can see a distant target more easily than the target can see the observer through the foliage. While a frustum model could potentially capture this asymmetry, it introduces additional computational complexity that was not feasible to reliably evaluate in this study.

Second, an expanding radius captures points further and further away from the true, central LoS vector. In dense environments, a wide cone at a long distance will inevitably be blocked by buildings or trees that do not physically obstruct the central line of sight, leading to unrealistic false occlusions.

Finally, an increasing cross-sectional area breaks the logic of fixed structural density thresholds. In our model, a solid obstacle is identified when the point count exceeds a threshold relative to the cross-sectional area. In a frustum model, a building located far from the observer (where the radius is large) might accumulate enough total points to trigger a "blocked" state, whereas that same building located close to the observer (where the radius is small) would not.

Ultimately, the cylindrical model used in this research is not sufficient to perfectly model human vision. However, it is the simpler choice for purely modeling the computational buffer necessary for calculating discrete intersections.

5.2 Attenuation

Attenuation can be approximated using several different attributes that are captured during or derived from LiDAR data. While density is an intuitive metric, other metrics could provide better performance.

Point density does not necessarily scale linearly with actual leaf density or attenuation. Trees are complex, and a higher density of leaves might actually decrease the captured point density internally or on the underside of the tree crown, as fewer laser pulses penetrate the outer foliage. Furthermore, point density and distribution are not determined solely by the physical terrain and objects being scanned. They are significantly influenced by the data acquisition parameters, including the arrangement of flight lines, scanning angles, and the degree of overlap between flight strips.

Because of these compounding variables, it remains challenging to definitively evaluate the performance of different attenuation metrics without comparing them against true attenuation values obtained through independent methods like photography.

5.3 Seasonal differences

Seasonal change entails more than just a change in leaf density, as vegetation is highly dynamic. It will not only become more dense during summer but also physically larger. Physical addition of random points partially replicates this effect, whereas increasing the attenuation constant does not. However, modifying the attenuation constant is probably preferable. Not only for simplicity and computational efficiency, but also because for most missions, underestimating visibility is much more problematic than overestimating it. Randomly adding points introduces additional errors that are likely not worth it, as we do not know exactly how far trees will expand in summer.

5.4 Facade enhancement

For capturing vertical facades, ALS is limited compared to TLS or pictometry methods. These other methods generally achieve better coverage of facades given their relative angle. Conversely, ALS is uniquely suited for analysing airborne missions as it closely matches the perspectives during data collection.

While the synthetic densification of facades proves useful against the issue of small LoS cylinders passing through sparse vertical data, it inherits some of the limitations of 2.5D models. It assumes that every detected roof polygon is supported by solid, opaque walls extending vertically down to the terrain. This ignores overhangs, archways, and other 3D building features. Crucially, the kind of error that these assumptions can introduce will cause an underestimation of visibility. However, given the increase in accuracy it achieves, the enhancement is probably worth its cost in most scenarios.

6. Conclusion

6.1 Main finding

This research demonstrates how direct point cloud analysis can overcome the key limitations of current methods by answering the central research question: *How can military aerial mission planning be improved by incorporating semi-transparent vegetation in probability-based visibility calculations computed directly on airborne LiDAR point clouds?* We established an efficient pipeline that enables rapid 3D Line of Sight (LoS) queries directly on large, remote ALS datasets. By leveraging Cloud Optimized Point Cloud (COPC) file streaming for targeted data selection, spatial indexing via k-d trees, and optimized cylinder-sphere step intersections, the proposed method bypasses traditional I/O bottlenecks and the need for intermediate representations.

In terms of accuracy and quality, this approach successfully integrates semi-transparent vegetation into a probabilistic model that generalizes across different datasets. We demonstrated that vegetation can be reliably segmented without manual intervention using the Myria3D deep learning model. By adapting the Beer-Lambert law, the algorithm utilizes point density to estimate visual permeability and realistically models visibility as an exponential attenuation. This non-binary approach improves accuracy compared to traditional binary models. Consequently, this model increases the hidden area in which missions can effectively operate while granting commanders the flexibility to implement precise, threshold-based risk management.

Furthermore, direct point cloud analysis outperforms industry-standard raster techniques given the same input data and output resolution. We found that synthesizing additional facade points mitigates inherent sparse facade scans, allowing for the use of thinner LoS cylinders to increase spatial resolution. The resulting high-fidelity 3D visibility spaces can be effectively visualized for mission planning through 3D GIS software, web viewers, or targeted 2D map overlays and cross-sections. Ultimately, transitioning to this non-binary, fully 3D attenuation model provides a performant, accurate, and flexible alternative to current workflows, enabling safer and more effective military airborne mission planning.

6.2 Future work

Challenges remain before this level of analysis is practical for most mission planning tasks. To move these techniques towards practical application, future work should focus on empirically determining the attenuation constant and an appropriate constant for compensating seasonal differences. Furthermore, attention must be given to point density control in the input data, as variations caused by the data collection method (such

as multiple laser returns and overlapping flight paths) can artificially inflate density and skew the attenuation calculations.

Additionally, LoS analysis should be integrated with other aspects of visibility such as glare, color, and environmental factors. To accurately represent the human field of view (FOV), future work should explore the use of viewing frustums or other models of human vision. Furthermore, the specific optical characteristics of different equipment, such as cameras and binoculars, could also be modeled. The method should also eventually be expanded to include other relevant operational aspects, such as acoustics and egress routes, for an integrated mission planning workflow. This workflow could potentially leverage Virtual Reality (VR) to visualize 3D viewsheds and routes for planners, and should account for dynamic, moving observers to analyze 3D routes over time.

Practical applications of these techniques could also be found in domains outside of military mission planning, such as urban planning, ecology, and telecommunications.

Future scientific work should focus on further optimizing the scalability and efficiency of direct point cloud processing, particularly to understand how analyzing significantly larger areas impacts performance. These aspects could be improved using advanced out-of-core processing, leveraging compiled languages, or optimizing for multi-core architectures. Moving towards a streaming computation model where data is continuously ingested could provide massive scaling benefits.

Another improvement could be the development of hierarchical methods. By dynamically evaluating thinned point clouds or high-level octree nodes, the algorithm could eliminate irrelevant geometries and request high-density chunks only for areas that are most relevant to the analysis. Specifically, future implementations could directly utilize the native octree structure embedded within Cloud Optimized Point Cloud (COPC) files for spatial selection, eliminating the computational overhead of constructing a separate k-d tree.

As LoS generation becomes more performant, I/O bottlenecks for exporting 3D results emerge as a limitation. Investigating space-efficient volumetric structures, such as Tetrahedral Irregular Networks (TINs), could optimize the storage of imbalanced 3D visibility spaces where much of the volume is empty. Additionally, results could be displayed not only as points or grids, but also as 3D surfaces, probability iso-planes, or Height Above Ground (HAG) surfaces.

Further point cloud processing techniques could further enrich the model by incorporating additional information. This could be done by including additional semantic classes like vehicles or specific tree species, or by detecting features such as gates, arches, and windows. Another addition would be to include temporal (4D) dynamics, such as using repeated scans and change detection to assign points a “speed” to estimate their uncertainty.

The core ideas of streaming, direct processing, and spatial indexing can be applied to other data sources. The most obvious of these are Structure from Motion (SfM) point clouds. An excellent candidate for future research is the open, nationwide 3D dataset provided by the Dutch Kadaster[6]. It generates point clouds from dense matching of high-resolution aerial images, yielding densities of up to 100 points/m². SfM generally offers more frequent updates and natively contains color channels to accurately classify vegetation. In return, it has a slightly lower vertical accuracy compared to traditional LiDAR, is dependent on daylight and weather conditions, and while it can classify vegetation well based on color, it specifically struggles to accurately capture its complex 3D geometry.

Finally, future work could explore other volumetric representations like 3D Gaussian Splatting (3DGS). Eventually, similar workflows could be developed for sources like radar and sonar.

[6] https://api.pdok.nl/kadaster/3d-basisvoorziening/ogc/v1/collections/digitaaloppervlaktemodel_8cm

3 Background

This chapter contains the background information required to contextualize the methodology proposed in this thesis. It begins by introducing military airborne mission planning to establish the operational context. Next, it defines the fundamental concepts of visibility analysis and the principles of Airborne Laser Scanning (ALS). An overview of the specific open-access LiDAR datasets and the Cloud Optimized Point Cloud (COPC) data structure used in this research is then provided. This is followed by an overview of point cloud processing frameworks and vegetation segmentation techniques. The theoretical foundation for probabilistic visibility, the Beer-Lambert attenuation law, is also detailed. Finally, the literature review places the research within the broader academic context of visibility analysis.

3.1 Airborne Mission planning

Airborne mission planning refers to the process of planning a (military) mission to be carried out using an airborne vehicle. Examples like UAV's and different helicopters are common. (Ministerie van Defensie, 2014) defines airborne vehicles as having one or more roles as presented in Figure 3.1. In these roles, airborne vehicles may carry out a wide variety of mission types.



Figure 3.1: Overview of the four roles of airpower
source: Ministerie van Defensie (2014, p. 30).

For almost all of these missions, 3D visibility is a crucial factor to consider during planning. Depending on the mission, different aspects are important. For some missions this could be remaining undetected at all costs, for others this could be searching a large area while minimizing time visible to the enemy. Generally, underestimating the visibility of an enemy observer is much more problematic than overestimating it, as it can lead to unexpected detection.

3.2 Visibility analysis

Line of sight

A line of sight (LoS) is a theoretical infinitely thin vector drawn between a source and a target point. It is used to determine if the visual path is obstructed by obstacles like the landscape, buildings, or vegetation. While a viewshed models visibility over a chosen area, a LoS provides a point-to-point calculation. When looking from a source point to a target point, the LoS can be either blocked, not blocked, or partially blocked in non-binary visibility models.

Viewsheds

Viewsheds extend the concept of a line of sight and show what part of the environment can be observed from a certain source. Conceptually, it is equal to computing the LoS from an observer point to every point in some area where every point is considered a target. Figure 3.2 shows viewsheds for two observers in some example area of terrain

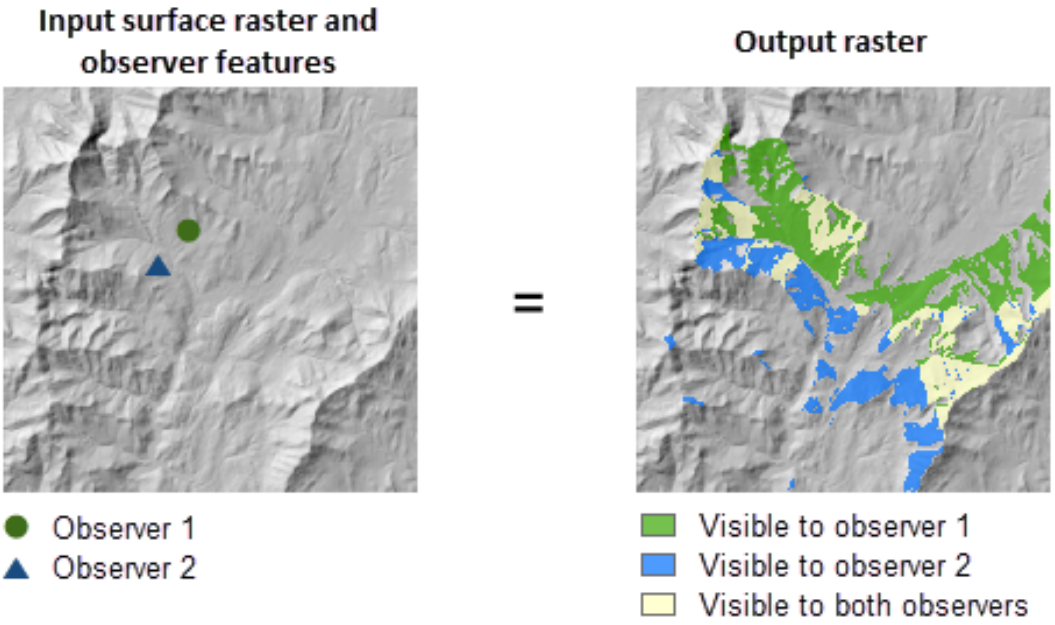


Figure 3.2: Example viewsheds of areas visible to two observers source: Esri[7]

Intervisibility

Intervisibility is the ability of two locations to be seen from one another. While a Line of Sight (LoS) refers to the specific line between two points, intervisibility describes the state or property of that connection. It is the fundamental building block of viewshed analysis, determining whether obstacles such as terrain, vegetation, or buildings permit a direct visual link between a source and a target.

Non-binary visibility

Traditional viewshed analysis uses a boolean value to describe visibility. An object is either "visible" or "not visible". However, visibility in the real world is often not so black and white, and a target might be "somewhat visible". Generally, this approach is called non-binary visibility. It acknowledges that visibility exists on a spectrum, influenced by environmental, physical, and probabilistic variables. Two similar but distinct terms are often used.

[7] https://pro.arcgis.com/en/pro-app/3.4/tool-reference/spatial-analyst/visibility.htm

Fuzzy visibility focuses on the clarity with which a target is visible, representing visibility on a spectrum from the perspective of the user. One might use decay functions to model factors that reduce clarity over distance or under varying weather conditions, such as fog, mist, or solar glare. A target might be categorized as "usually visible" or "visible only under very favorable conditions" depending on different factors.

Probable visibility focuses on the likelihood that an unobstructed line of sight exists between an observer and a target. This model accounts for uncertainties and physical obstructions that are difficult to map with absolute precision, such as Digital Elevation Model (DEM) errors or the presence of vegetation. Since it is computationally infeasible to model every individual leaf and branch in a forest, one might use a probabilistic approach to determine the chance that LOS beams will successfully pass through such layers.

3.3 Laser scanning

Many spatial analyses are based on LiDAR data using either Airborne Laser Scanning (ALS) or Terrestrial Laser Scanning (TLS) techniques. The high degree of scalability and automation, together with continuously improving accuracy and pricing of hardware, make them accurate and cost-effective techniques for capturing spatial data (Pyysalo et al., 2009).

While exact figures diverge, the global and military market for LiDAR technology is growing and expected to grow further in the coming years (industryarc, 2025; Sharma, 2026). One example of this is more and more countries capturing and releasing ALS point cloud datasets as open data. The Netherlands[8], France[9], Spain[10], Canada[11], and the United States[12] among others provide current and historical dataset.

ALS is a highly efficient method that enables large areas to be captured and provides high-resolution 3D data (Pyysalo et al., 2009). The core technology is a LiDAR scanner mounted to an airborne platform like a plane, helicopter, or UAV. When the captured LiDAR data is processed together with information gathered by a GNSS receiver and an Inertial Measurement Unit (IMU), a single geolocated point cloud is produced. Figure 3.3 shows a diagram with all the main systems used for ALS.

[8] https://basisdata.nl/hwh-portal/download/index.html
[9] https://cartes.gouv.fr/telechargement/IGNF_NUAGES-DE-POINTS-LIDAR-HD
[10] https://centrodedescargas.cnig.es/CentroDescargas/lidar-tercera-cobertura
[11] https://open.canada.ca/data/en/dataset/7069387e-9986-4297-9f55-0288e9676947/resource/4570da77-421b-4cf0-880e-cd475f570245
[12] https://apps.nationalmap.gov/lidar-explorer/#/

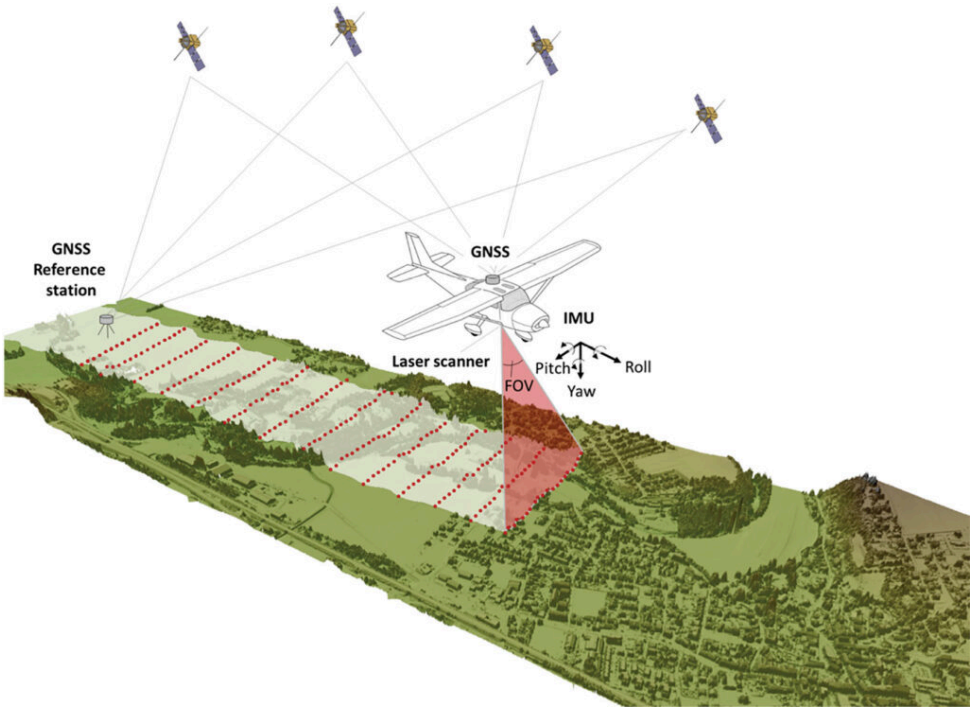


Figure 3.3: Diagram showing the process of Airborne Laser Scanning (ALS) source: LiDARmagazine[13]

One interesting consequence of ALS is the fact that the data is collected from the perspective of the chosen platform. Nebiker et al. (2010) points out that "In order to obtain detail-rich reality-based 3D environments, data acquisition needs to take place from a similar perspective of the envisaged users". For some applications like visibility modeling for pedestrians, this difference in perspective can be problematic when choosing ALS over TLS. However, for the application proposed in this thesis, this argument flips because the perspective from which the data was captured matches the perspective of analysis almost exactly.

An unavoidable issue in laser scanning data is that of occlusion. This is where part of the captured area is blocked by other geometry like buildings, resulting in gaps in the final product. Proven methods that deal with occlusion in point clouds may not be as suitable as one might think in the case of military applications. Often, assumptions are made regarding the symmetry of buildings (Oude Elberink, 2010, p. 166), yet this symmetry is often disrupted in disaster or war zones. Other methods, such as those used by the 3DBAG[14], utilize additional data like footprints or maps for reconstruction. These assume that footprints remain relatively stable, but in this context, that is no longer a given. This is especially important given that overestimating concealment is often far more dangerous than underestimating it.

Additionally, direct point cloud analysis is not as widespread as classic raster and vector-based methods and is thus not as widely supported (Nebiker et al., 2010). Another challenge is the large storage requirements for point cloud datasets when compared to derived representations. In the past ten years, much work has been done on developing and implementing techniques for efficiently storing, clustering, and visualizing point clouds. Developments like Cloud Optimized Point Clouds (COPC) together with increasingly powerful hardware have resulted in direct analysis becoming more practical and more widely used each passing year.

[13] https://lidarmag.com/2024/12/30/airborne-lidar-a-tutorial-for-2025/
[14] https://docs.3dbag.nl/en/

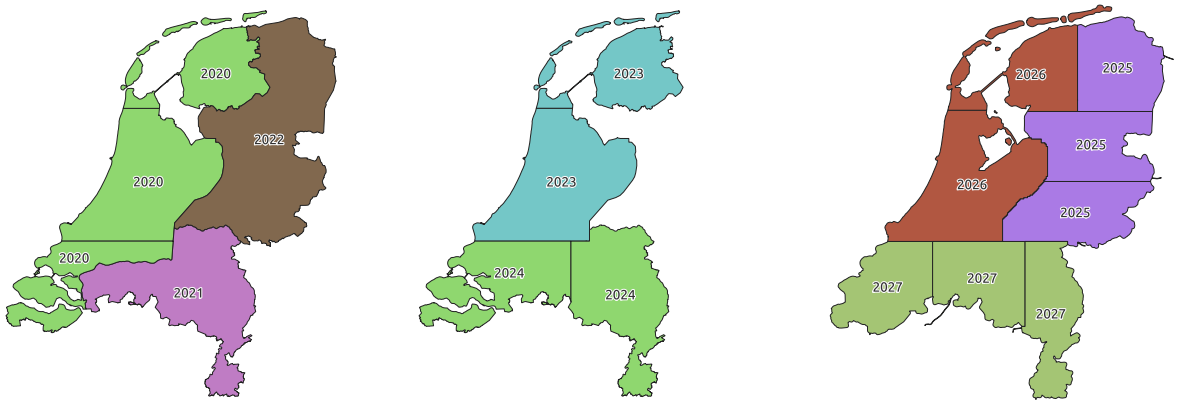
3.4 Data

This section provides an overview of the data sources utilized throughout this research to develop, test, and validate the proposed visibility algorithms. It begins by introducing the three primary national LiDAR datasets: the Dutch AHN, the French LidarHD, and the Canadian CanElevation. Then, these datasets are systematically compared in terms of their included LiDAR attributes (Table 2) and their broader acquisition specifications, such as density and seasonality (Table 3).

It is important to note the existence of other classified data available to the Dutch Ministry of Defense and its allies. These data are maintained by the 'Dienst Geografie' (Dienst Geografie, 2024) and may have different characteristics from the open datasets used. However, restricted datasets are excluded from this thesis to strictly adhere to Open Science principles. A core objective of the work is to ensure that the developed method is reproducible and verifiable by reviewers and other researchers. Reliance on classified data would prevent the open publication of the code and dataset. The discussed public datasets serve as a sufficiently high-quality source for transparently developing these algorithms.

AHN

The 'Algemeen Hoogtebestand Nederland (AHN)' or general height model of the Netherlands is a multi-year collaboration between local and national government organizations. Each version, except AHN 5 and 6, maps the entire country using helicopter or plane-based LiDAR scanning and is released as open data[15]. Figure 3.4 shows when the three most recent versions were collected for different parts of the Netherlands.



(a) AHN 4 (b) AHN 5 (c) AHN 6

Figure 3.4: Year of capture of AHN 4, 5, and 6 source: AHN[16]

IGNLidar HD

The French 'LiDAR HD' project is a mapping initiative similar to the AHN by the French National Institute of Geographic and Forest Information (IGN). Launched in 2021, the program aims to scan the entirety of France and its overseas departments using airborne LiDAR by 2026[17].

[15] https://www.ahn.nl/ahn-the-making-of
[16] https://www.ahn.nl/dataroom
[17] https://geoservices.ign.fr/lidarhd

Canelevation

The 'CanElevation' LiDAR Point Clouds product supports the National Elevation Data Strategy implemented by Natural Resources Canada (NRCan). Unlike the uniform national sweeps of the AHN and IGN projects, CanElevation aggregates airborne LiDAR data from various independent projects conducted by NRCan alongside federal, provincial, territorial, and municipal partners. Due to Canada's vast size, high-resolution LiDAR acquisitions are primarily focused south of the productive forest line.

Attributes

The attributes present in LiDAR point clouds generally follow the LAS 1.4 standard. While providers reliably include the core attributes of the standard, there are significant differences between datasets and versions when it comes to the extended attributes. Table 2 provides an overview of available attributes in each dataset.

Attribute	Description	AHN	LidarHD	CanElevation
Core ASPRS LAS 1.4 Attributes				
Coordinates	X, Y, and Z spatial coordinates	✓	✓	✓
Intensity	Reflective strength of the laser pulse	✓	✓	✓
Returns	Return number and total number of returns	✓	✓	✓
Angle	Scanning angle of the sensor	✓	✓	✓
Timestamp	Absolute GPS time of capture	✓	✓	✓
Classification	Base category label (e.g., ground, building)	✓ no vegetation	✓ including vegtation	✓ inconsistent vegetation
Extended ASPRS LAS 1.4 Attributes				
RGB Color	Color from simultaneous aerial photography	✓*	×	~
Flightpath ID	Point Source ID / Flight line identifier	✓	✓	×
Scanner Channel	Channel of the sensor for multi-channel systems	×	✓	×
Scan Direction	Direction of the scanning mirror	×	✓	×
Class. Flags	Advanced binary flags (e.g., Synthetic, Overlap)	×	✓	×

Table 2: Comparison of available attributes across the three national datasets.
* AHN 1-5 colorized using satellite imagery

AHN is classified in a semi-automated manner. After outlier filtering the point cloud is automatically classified and corrected by hand[15]. The available classes are: `terrain`, `buildings`, `infrastructure`, `power lines`, `water`, and `other` which includes vegetation.

Similar to AHN, Lidar HD is classified using a semi-automated pipeline that mixes classical and learning-based algorithms. Crucially, vegetation is explicitly classified and separated into three distinct height classes: low, medium, and high.

Canelevation classes depend on the specific acquisition project. For instance, points on water surfaces might be explicitly classified as `water` if requested by the project specifications, or simply grouped into `ground` if they were not.

Dataset specifications

The selected datasets are all available under public licenses that make the data suitable for scientific research and publication.

They are all available in COPC with the exception of the color-enriched geotiles distribution of AHN 1-5.

Point density is generally around 10 pts/m², but there is some variance between versions and a lot of variance in CanElevation.

	AHN	**LidarHD**	**CanElevation**
License	CC-0 1.0 (AHN 2-4) CC BY 4.0 (AHN 5,6)	French Open Licence 2.0 (Licence Ouverte)	Open Government Licence - Canada
File Format	LAZ (geotiles[18]) COPC (basisdata[19])	COPC[20]	COPC
Point density [pts/ m²]	6-10 (AHN 2,3) 10-14 (AHN 4-6)	>10 (below 3200m) >5 (above 3200m)	–
Season	winter	year-round	year-round

Table 3: Comparison of specifications across the three national datasets.

To maximize ground resolution, AHN is collected exclusively during the so-called "leaf-off" season. This could be cause for concern when investigating vegetation, as leaves are the most significant contributor to their visual obstruction. However, as shown in Figure 3.5, even during leaf-off season, this does not mean there are no leaves at all.

[18] https://geotiles.citg.tudelft.nl/
[19] https://basisdata.nl/hwh-portal/download/index.html
[20] https://cartes.gouv.fr/telechargement/IGNF_NUAGES-DE-POINTS-LIDAR-HD



Figure 3.5: AHN6 point cloud sample visualized in CloudCompare

Furthermore, as vegetation is inherently dynamic, every measurement provides only a snapshot of its state. When collected out of season, leaf-off measurements underestimate the impact of leaves, whereas leaf-on measurements overestimate it. This means any practical implementation that does not rely on highly recent data must eventually provide some way to incorporate or compensate for seasonal changes regardless of when the data is captured.

LidarHD and CanElevation do not have specific restrictions for when data is collected but will encounter the same issue when the moment of collection falls inside a different season than the moment of use.

3.5 COPC

The Cloud Optimized Point Cloud (COPC) format is a relatively new standard for storing point cloud data. A COPC file is a valid, standard compressed LAZ 1.4 file, but with extra information added for optimized querying. This means any legacy LiDAR processing software or viewer capable of reading a standard LAZ file can open and read a COPC file. However, COPC-aware software like the PDAL library and newer versions of CloudCompare can use its efficient streaming capabilities.

COPC enables highly efficient remote streaming via standard HTTP Range requests. This means that when a user is only interested in part of a file, which is extremely common, only that part can be streamed to their machine. COPC achieved this by organizing the points into a hierarchical octree structure, which can then be queried to skip to the relevant part of the file immediately without having to waste storage, bandwidth, and decompression time on the rest of the file.

Figure 3.6 illustrates how this spatial index is encoded in the COPC file. Following the standard LAZ `HEADER`, the point data is grouped into chunks. The key COPC addition is the `COPC EVLR` at the end of the file. This EVLR stores the spatial index, mapping the octree hierarchy to the exact byte offsets of the chunks of points. When a request by software that supports COPC is made, it first reads the EVLR, jumps to the exact byte where the relevant chunks are stored, gets those using a HTTPS range request, and decompresses them according to the LAZ standard.

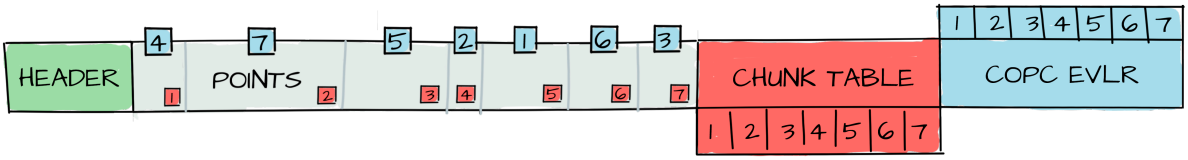


Figure 3.6: Diagram showing the layout of a COPC file
source: COPC.io[21]

Because the method proposed in Section 2.3 uses a spatial index, a natural idea would be to reuse the octree included in the COPC file. Unfortunately, current tools like PDAL do not yet support this reuse paradigm.

3.6 Point cloud processing libraries

For efficiently processing large LiDAR datasets, four frameworks were considered. All of them can be used with Python and are designed to process point clouds. All four are briefly discussed below, after which Table 4 compares the different frameworks by ranking them on a number of criteria.

- **PDAL** is a flexible C++-based framework used for the processing of (LiDAR) point cloud data through a modular pipeline architecture. It offers deep integration for hybrid Python/C++ development, making it possible to prototype logic in Python while performing performance-critical calculations using custom C++ filters. PDAL's COPC reader is highly flexible and supports complex spatial queries via Well-Known Text (WKT) polygons.
- **LAStools** is an industry-standard suite of modular C++ binaries optimized for rapid I/O operations and low-memory overhead. The framework operates on a dual-license model where the foundational libraries (LASlib and LASzip) are open-source and high-level processing tools are proprietary. Although LAStools provides the highest raw processing speeds for local datasets, its use of standalone executables and its optimization for Windows make integration on Linux(servers) and Mac OS more difficult.
- **Laspy** is the most well-known Python library for handling LAS/LAZ formatted point clouds. Since version 2.2, it also provides support for COPC, enabling efficient spatial queries from remote HTTP servers. While it is very user-friendly and integrates well with other Python libraries, it lacks the modular pipeline architecture found in more specialized frameworks.
- **WhiteboxTools** is a high-performance set of backend tools written in the Rust programming language, focusing on multi-threaded geospatial processing and analysis. It uses an open-core model where most of its tools are MIT-licensed, and advanced classification features are reserved for a proprietary extension.

[21] https://copc.io/

Feature	PDAL[22]	LAStools[23]	laspy[24]	Whitebox[25]
Open source	✓✓✓	✓	✓✓✓	✓
COPC Streaming	✓✓✓	✓✓	✓	✓
3D Flexibility	✓✓✓	✓✓✓	✓✓	✓
C++ Integration	✓✓✓	✓✓	×	×
Ease of use	✓✓	✓	✓✓✓	✓✓✓
General performance	✓✓	✓✓✓	✓	✓✓✓
OS Support	✓✓✓	✓	✓✓✓	✓✓✓

Table 4: Comparison of different frameworks for point cloud processing ranked from not suitable(×) to most suitable(✓✓✓)

Based on the comparison, this thesis uses PDAL as the primary processing framework. Its superior polygon COPC streaming, open-source philosophy, and integrated Python/C++ pipelines make it the most suitable tool.

3.7 Segmenting vegetation in point clouds:

Automatically segmenting vegetation in point clouds is a widely studied problem, and there exists several software for this task. These are both open source, like the CANUPO[26] plugin in the Cloud Compare software, as well as closed source, like the ArcGIS tree point classifier[27]. These software packages are not preferred for the method proposed in this thesis since they are harder to integrate into an automated pipeline and require the user to install the specific software on their machine.

Another common approach is rule-based classification of point clouds. These methods will generally use properties like point height, neighbor distance, and local planarity to detect trees, like the LAStools lasclassify[28] tool. Open-source alternatives that implement rule-based classification exist, like a segmentation algorithm developed by the Amsterdam AI team[29]

Whitebox's flash point classifier[30] implements the same functionality but is part of their proprietary set of tools, meaning it is not possible to determine their exact methodology and is thus not suitable for this thesis.

[22] https://pdal.org/en/2.9.3/
[23] https://lastools.github.io/
[24] https://laspy.readthedocs.io/en/latest/
[25] https://www.whiteboxgeo.com/
[26] https://www.cloudcompare.org/doc/wiki/index.php/CANUPO_(plugin)
[27] https://www.arcgis.com/home/item.html?id=58d77b24469d4f30b5f68973deb65599
[28] https://lastools.github.io/download/lasclassify_README.md
[29] https://github.com/Amsterdam-AI-Team/Tree_Detection_in_Aerial_Point_Clouds?tab=readme-ov-file
[30] https://www.wgsoftware.xyz/

Other, mostly newer, methods rely on deep-learning algorithms to classify vegetation in point clouds. Companies like Pointly[31] offer an API that implements such a method, but their algorithm is proprietary. Pointtree[32] offers an open-source state-of-the-art implementation but is not meant to be used for urban regions, which makes it less suitable.

Myria3D (Gaydon, 2022) performs "Aerial LiDAR HD Semantic Segmentation with Deep Learning", which is a cutting-edge approach to classifying vegetation in point clouds. It is also open source, has good documentation, and is being maintained by the French Geographic Institute IGNF (Institut National de l'Information Géographique et Forestière). These qualities make it the most suitable candidate for the proposed method.

3.8 Beer-Lambert attenuation law

When first introducing his visual permeability method, Dean (1997) uses a linear model, which means each additional tree that blocks the LoS contributes the same amount to the total probability of a target being visible. Llobera (2007) challenges and improves this model, for which he uses a theoretical basis in physics, the Beer-Lambert attenuation law.

The Beer-Lambert law describes how light intensity drops as it passes through a particle-filled slice of material. In his analogy, light intensity is the probability of seeing our target, the beam of light is the LoS, and the particles are leaves on trees.

The Beer-Lambert law is given in Equation 3.1, which relates the output intensity I to the input intensity I_0. This depends on the attenuation coefficient μ and the thickness of the material x. This means that increasing either the attenuation coefficient or the thickness will exponentially reduce the output intensity.

$$I = I_0 \cdot e^{-\mu \cdot x} \tag{3.1}$$

Llobera (2007) derives Equation 3.2 where the probability of seeing a target $p(x)$ is calculated using a function $k(x)$ based on the density of vegetation along the LOS and the length of the LoS passing through the trees x.

$$p(x) = e^{-k(x) \cdot x} (k \geq 0, x > 0) \tag{3.2}$$

The formula assumes a homogeneous composition of particles. While this holds in the theoretical limit as x approaches zero, discrete implementations must approximate this continuous attenuation by evaluating the vegetation density in discrete steps along the LoS.

Previous methods assumed visibility decreased linearly when obstructions were found (Dean, 1997). The key innovation in this method is that the probability of seeing a target decreases exponentially as the distance into the vegetation increases.

3.9 Literature review

Method for finding sources

For a complete and transparent summary of related work, a small literature review was performed based on the PRISMA protocol (Page et al., 2021). Searches were done in the Scopus[33], The Lens[34], and Dimensions[35] databases. The full keyword queries are available in

[31] https://pointly.ai/#:~:text=and%20Analysis.,Platform%20for%20Point%20Cloud%20Management
[32] https://github.com/ai4trees/pointtree?tab=readme-ov-file
[33] https://www.scopus.com/
[34] https://www.lens.org/
[35] https://www.dimensions.ai/

Appendix C and consist roughly of three parts customized to the syntax of the individual database. Each result must contain a mention, synonym, or term closely related to 1) visibility analysis, 2) non-binary visibility, and 3) LiDAR.

In the final review, Dimensions contributed zero papers ($n = 0$) using the selected keywords. While Scopus and The Lens allow targeted searches across a combined "Title, Abstract, and Author Keywords" field, Dimensions lacks a direct equivalent to an author-keyword search. Instead, it relies on AI-driven concept extraction and full-text indexing (Hook et al., 2018). Limiting the query to titles and abstracts in Dimensions returned no results. Expanding the search to the full text returned 1,321 results, which was too many to screen manually.

These papers identified from Scopus and the Lens were used as a starting point for snowball-style search, complemented by supervisor recommendations and the connected papers[36] tools. The body of collected work was sorted and filtered by hand using Rayyan[37]. Figure 3.7 shows a PRISMA-style flowchart of the origin and number of sources reviewed.

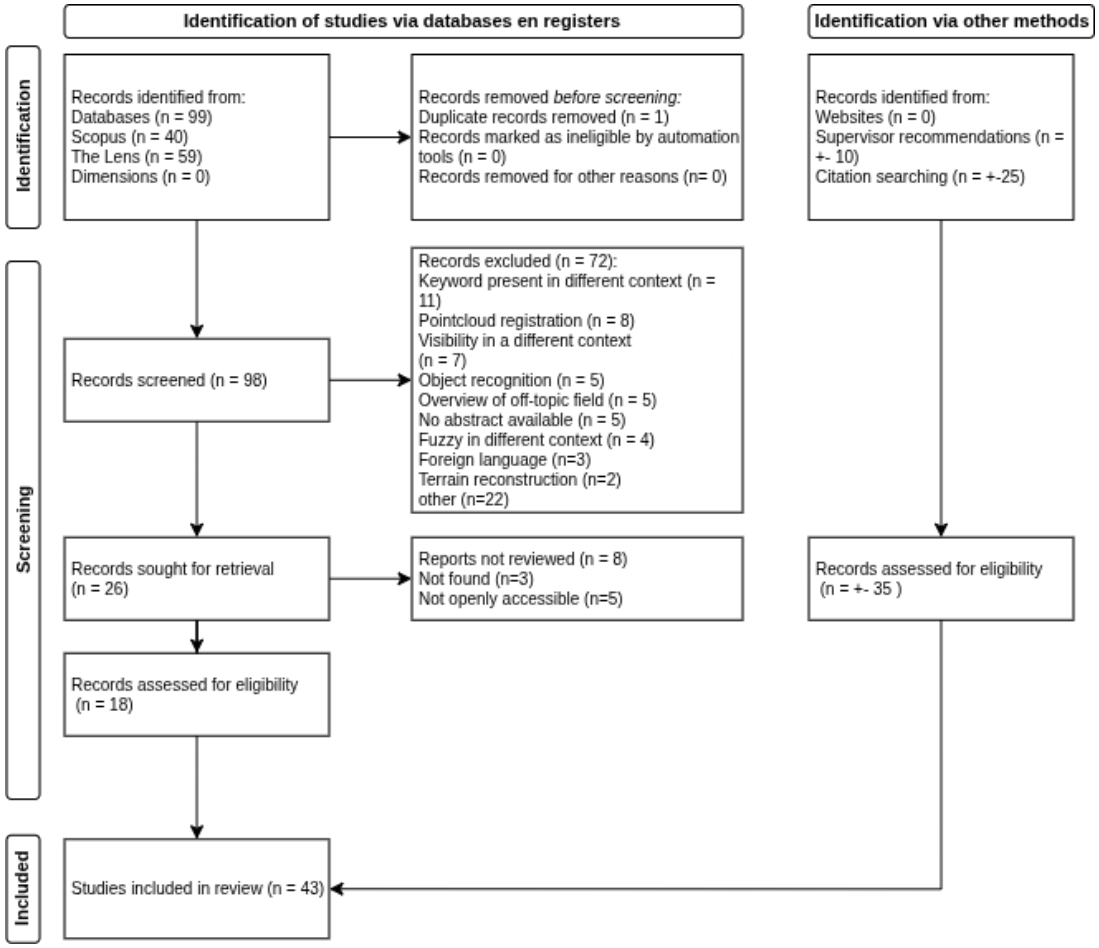


Figure 3.7: Flowchart showing how sources were filtered and selected as a result of both systematic searching for combinations of keywords and reference searching.

The remainder of this section reviews the evolution and current state of visibility analysis across five key themes. First, the treatment of **vegetation** is examined, tracking its progression from being entirely excluded to being modeled as an interactive environment variable. This leads directly into a discussion on **non-binary visibility**, detailing how mathematical models

[36] https://www.connectedpapers.com/
[37] https://www.rayyan.ai/

like visual permeability and the Beer-Lambert attenuation law have been applied to calculate probabilistic sightlines. Next, the **dimensionality** of visibility models is explored, discussing the shift from 2.5D surface representations to fully 3D environments. Then, various spatial **representations** like voxel grids, meshes, and point clouds are compared in terms of their capabilities and limitations. Finally, the challenge of **visualization** is covered by reviewing methods for effectively communicating complex 3D visibility data to end-users.

Vegetation

Guth (2009) provides a key starting point in incorporating vegetation in visibility analysis. Their method achieves this using two raster grids, a DTM and a DSM, derived from LiDAR measurements. This early implementation approximates the included vegetation as fully opaque.

Bartie et al. (2011) states that vegetation is often excluded and includes it using a separate raster called a canopy height model or CHM. They mention some of the issues with current methods like semi-transparency and 2.5D representations. Their results indicate that including vegetation makes the analysis both more stable and more accurate. They also include a binary mask for deciduous and evergreen trees where deciduous trees are set to a single minimal visual permeability during winter. One limitation of their method is the requirement for manual surveying and reliance on somewhat subjective averages for visual permeability.

Murgoitio et al. (2013) Includes trees by focusing only on tree trunks, excluding leaves or branches. They use a raster as an intermediate representation and calculate full viewsheds using the ArcGIS software.

Maaiveld et al. (2024) offer the first implementation including vegetation in a non-binary visibility algorithm specific to military mission planning. Their work applies this method to autonomous ground vehicles and path optimization. Like most, they first convert the raw point clouds to a grid, in this case hexagonal. Their model for attenuation approximates the effect as a linear combination using fixed obstruction weights for trees and bushes. When calculating LoS, the combined visibility is based on the minimum transparency along the LoS, which limits how the combination of multiple blocking trees interacts.

Non-binary visibility

Dean (1997) establishes the first method that models visibility where vegetation is represented as a semi-transparent obstacle. Dean identifies 3 problems of raster-based methods: LoS are rasterized causing error: obstructions are evaluated only at raster cell centers: and vegetation is modeled as either fully opaque or fully transparent. They then introduce a method based on TINs and a technique called the visual permeability method. The proposed method starts with a visibility value of 1 (totally visible) and decreases it each time an obstacle is encountered based on its visual permeability and the length of the LoS through the obstacle.

Given that the method is based on TINs, it remains limited to 2.5D representation. Additionally, the visual permeability based on any measurements and the way in which it is defined does not consider the exponential nature of cumulative visibility. Llobera (2007) builds on Dean (1997) and proposes an algorithm to develop a more accurate cumulative visibility for LoS. The raster-based method models the exponential decay in visibility based on Beer-Lambert's attenuation law.

Govindaraju et al. (2014) reiterates that current methods omit or overestimate partial obstruction, specifically vegetation. Their method assumes a constant mean vegetation height

and uses tree cover as a measure of vegetation density. Similar to Llobera (2007), they use the Beer-Lambert attenuation law for their model. Their algorithm includes terms for complete occlusion (terrain), partial occlusion (vegetation), and the sensing ability of the sensor. Their algorithm uses the fast marching method for determining visible points, which works by comparing the geodesic distance (including obstacles) to the Euclidean distance.

Most recently, Leenders et al. (2025) map the larger problem in their scoping literature review. They provide a systematic overview of different aspects of fuzzy and probable viewsheds specifically and mention military helicopter mission planning as an application for these techniques.

Methods that incorporate a non-binary vegetation model have used a variety of metrics to approximate transparency. For instance, Bartie et al. (2011) uses ground perspective images to evaluate permeability by hand, but suggests side-looking LiDAR as a future automated approach.

Guth (2012) divides the LOS into small bins and counts returns within vertical voxels to derive vegetation density at various heights.

Ruzickova et al. (2021) uses a classifier to compute the ratio of sky vs leafs on pictures per tree species

Mistick et al. (2023) computes a visibility index from DSM-based viewsheds at sampled locations. They then train a Random Forests classifier using 146 terrain- and vegetation-based metrics at 10 m resolution, many of which are LiDAR-derived vegetation structure measures.

Finally, Zong et al. (2023) use TLS and derived metrics to train a random forest model including 10th height percentiles of canopy returns, normalized relative point density (NRD), and the coefficient of variation of 99th height percentiles of understory returns.

Dimensionality

An overwhelming majority of methods reviewed use a 2.5D representation to model visibility (Bartie et al., 2011; Govindaraju et al., 2014; Guth, 2009; Llobera, 2007; Maaiveld et al., 2024; Mistick et al., 2023; Murgoitio et al., 2013). Depending on the method, this is done through a combination of a Digital Elevation Model (DEM), Digital Surface Model (DSM), Digital Terrain Model (DTM), and sometimes a Canopy Height Model (CHM). These are represented as square or rectangular rasters, with the exception of Maaiveld et al. (2024), who use hexagons.

Orlof et al. (2024) shows that representing terrain as a Triangulated Irregular Network (TIN) extracted from point clouds offers improved accuracy compared to DTMs. However, TINs still only accept one height value for each planar position and are thus 2.5D, meaning overhangs cannot be represented.

Zhao et al. (2024) propose a framework to categorize the dimensionality of different methods. They conclude "True 3D requires multiple viewpoints and the fusion of multiple depth maps, which can only be processed using point clouds."

Representation

Alsadik et al. (2014) describes three general approaches for viewing point clouds in a computerized 3D environment: the triangulation approach, where a set of connected triangles form a mesh; the voxel approach, where small cubes are visualized; and the Hidden Point Removal (HPR) approach, where the point cloud is viewed directly. Here, this framework is

used to discuss a number of representations that fall into these categories and extend it with some that do not.

Meshes convert the point clouds to a network of triangles using one of several available algorithms, like ball pivoting (Bernardini et al., 1999) or Poisson reconstruction (Kazhdan et al., 2006). This intermediate representation can then be used to perform surface normal testing and z-buffering to determine visible points. Besides the approximation required, one limitation is the difficulty of representing not only the shell of an object but also the interior volume. This involves constraints on topology and watertightness, which are non-trivial issues.

Voxel grids convert the point cloud to voxels of some set resolution first. Then, ray-voxel intersection and ray-tracing are performed Díaz-Vilariño et al. (2018) to determine visible points.

(Pyysalo et al., 2009) uses three voxel-based models of classified ALS data to compare a DEM, DSM, and combined model. In their method, every voxel that contains even a single point is considered full. In their testing, the DSM and combined model show little difference because of this binary approach, while both clearly outperform the DEM. They use average intensity per voxel as a visualization technique and mention using it when determining the value of voxels in their future work section.

After going through the limitations of 2.5D representations, Fisher-Gewirtzman et al. (2013) uses a fully 3D voxel-based approach for analyzing visibility in urban environments. Their method requires the user to input a CAD model of the relevant buildings and set a number op parameters like voxel resolution. A recurring issue clearly discussed in their paper is implementing rules or workarounds for cells that are on the boundaries of buildings.

Díaz-Vilariño et al. (2018) uses a voxel structure where each voxel is labelled as either exterior, occupied, visible, or occluded based on an occupancy algorithm. Once they establish this representation, visibility analysis is performed using a ray-tracing algorithm. They use this approach to calculate 3D isovists from the boundary of visible voxels for a given observer position.

Zong et al. (2021) systematically examine the impact of voxel size, forest type, and understory cover on visibility analysis results. They find voxel size to have a strong effect on accuracy, with an optimum at around 10cm resolution in their testing. However, they mention this optimal resolution can vary, creating a balancing act between having few unmapped voxels and high levels of detail. In their testing, the optimal resolution also varies based on understory cover of the selected area.

While more suitable for representing volumetric objects, voxel models introduce some complex issues. Oftentimes, the user is required to set the resolution, direction, and origin of the 3D grid, like in Fisher-Gewirtzman et al. (2013). Zong et al. (2021) shows this is a hard problem where there are no easy ways to predict the effects on accuracy beforehand. Additionally, every voxel-based model, in its own way, must deal with special cases where voxel boundaries (partially) intersect the boundary of the modeled objects, as shown by Fisher-Gewirtzman et al. (2013).

Point clouds can also be visualized directly. Introduced by Katz et al. (2007), the Hidden Point Removal (HPR) directly processes input point clouds by transforming the cloud using spherical flipping, then computing the convex hull to determine visible points.

Mehra et al. (2010) rigorously analyze how noise affects the convex hull in the inverted domain and derive theoretical bounds (guard zones) for visible point displacement. They introduce

a robust HPR operator that, instead of marking only points on the convex hull as visible, also considers points within a margin, based on the maximum expected noise.

Katz & Tal (2025) improve their own HPR method into HPRO (optimized) by introducing a method that identifies extreme points via a differentiable criterion, enabling gradient-based optimization like those in many machine learning pipelines.

Some of the latest research on the HPR method tries to improve on HPR by learning intervisibility using deep learning models (Wang et al., 2025).

Nebiker et al. (2010) introduce semantically rich point clouds as a new paradigm for (city) modeling and an alternative to geometric (mesh) 3D-modeling and image-based modeling methods. They argue that point cloud-based methods are better for micro- to meso-scale environments because of their high accuracy and level of detail. They are faster to produce and update in a fully automatic manner and so provide a simpler and cheaper alternative.

Poux et al. (2016) note that while point cloud acquisition techniques are now widely used, the resulting data is still underutilized. Similar to Nebiker et al. (2010), they argue that to fully use these data, they should be enriched by adding things like semantic information, structure, and color. They propose the Smart Point Cloud (SPC) as a data structure that integrates point cloud properties with human knowledge through machine learning to extract "smart data" rather than just raw points. They describe a framework to create SPCs by combining device expertise, analytical expertise, and domain expertise.

Guth (2012) implements an early version of a probabilistic intervisibility algorithm taking into account vegetation directly on point clouds, building on the work of Guth (2009). They show that point clouds offer superior accuracy when compared to raster-based approaches.

Zhang et al. (2021) study visibility analysis by directly processing point clouds. They show that taking into account (opaque) vegetation in visibility calculations directly on point clouds is possible. They specifically mention semi-transparent vegetation as a limitation and propose investigating its influence for future work.

Raw point cloud representations simplify some aspects of visibility analysis by removing the need for an intermediate representation requiring extra time, effort, and processing. At the same time, they introduce other forms of complexity, the most obvious one being the large amounts of data used to achieve their high accuracy and level of detail.

Stacked columns are another option, like the implementation by Lu & Han (2020). This involves turning buildings into stacked columns and then storing the height information and coordinates of each in a vector database. They show impressive results, reducing storage requirements and computation time. To realize this compression logically means that information is lost depending on the resolution.

The medial axis transform is also used to represent point clouds, as in the methods introduced by Peters et al. (2015). They overcome the limitation of the HPR method, which can only determine the visibility of points that are part of the point cloud, not of any arbitrary point. However, in order to generate the MAT, their method requires the point cloud to have normals, which are very chaotic for vegetation and trees specifically.

Machine learning has recently been used in methods that represent the full viewshed by sampling a subset of cells, relying on learning-based techniques to achieve full coverage. Mistick et al. (2023) uses a random forest machine learning model to predict VI (percentage

visibility in viewshed) for every pixel by training on a calculated subset of viewsheds. They note vegetation features had a larger impact on predictions than terrain features for their models. Their method offers an efficient way to predict full viewsheds while only calculating the visibility for a small subset of the total area.

Visualization

When tasked with the visualization of a 3D visibility analysis, one might naturally try to represent the results as a 2D view of the full 3D result. Pyysalo et al. (2009) conclude that the main challenge in this approach is information overload caused by voxels obscuring each other. They use adaptive transparency and assign 3D voxels a transparency value based on intensity, where high intensity values are considered opaque and low intensity values transparent, as shown in Figure 3.8. They also use symbols smaller than the voxel resolution to visualize what lies behind the top layer, as shown in Figure 3.9.



Figure 3.8: Captured image of the intensity visualization. Each voxel is assigned a transparency value based on its average intensity. The model has been made by Christopher Wood
source: (Pyysalo et al., 2009)

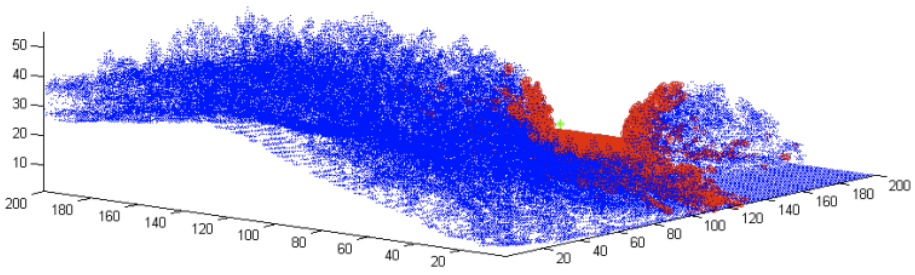


Figure 3.9: Example of the symbol visualization. The full voxels are represented as blue dots, which are surrounded by the red circle in cases where they are visible from the green asterisk.
source: (Pyysalo et al., 2009)

Díaz-Vilariño et al. (2018) use a 3D surface-based visualization enclosing the volume of the space view (a), and a voxel visualization using color (b) to visualize 3D isovists shown in Figure 3.10

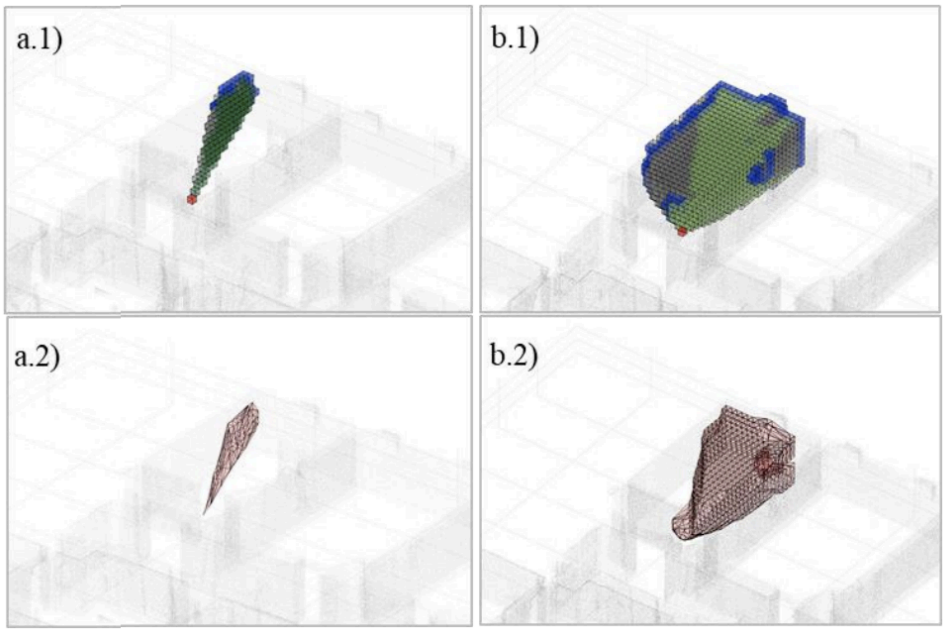


Figure 3.10: Diagram showing 3D visualization accomplished from two perspectives: a surface-based representation enclosing the 3D isovist (a), and a voxel representation (b).
source: (Díaz-Vilariño et al., 2018)

4 Reflection

This chapter reflects on the research conducted, its impact, and the thesis as a whole. It begins with an extended discussion exploring the mathematical and practical nuances of the proposed methods in Section 4.1. Next, Section 4.2 explicitly answers the primary and secondary research questions laid out in the introduction. Following this, the main contributions of the research to both military operations and the broader scientific community are discussed in Section 4.3. Then, the chapter reviews the modular software design process in Section 4.4. Finally, Section 4.5 evaluates the project's overall success against its initial MoSCoW goals and the core learning objectives of the MSc Geomatics program.

4.1 Extended discussion

Mathematical model for optimizing query radius

This section expands on the optimization discussed in Section 2.3. Given Figure 4.1 and Equation 4.1, it mathematically formalizes the volumetric trade-off between radial waste and query overlap.

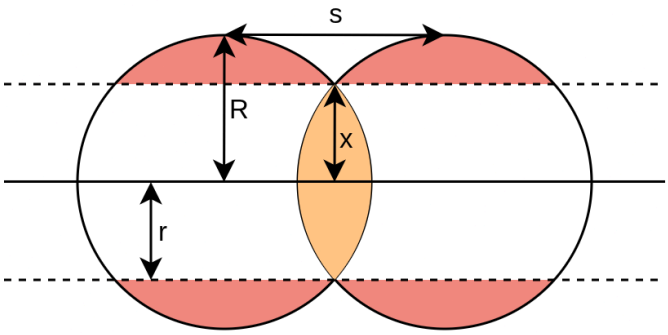


Figure 4.1: 2D diagram of the cylindrical volumetric line of sight model.

$$R = \sqrt{\frac{s^2}{4} + r^2} \tag{4.1}$$

Equation 4.1 shows there exists an infinite set of geometrically optimal solutions that completely cover the cylinder while minimizing radial waste. However, the choice of the step size s (and consequently the query radius R) drastically affects the computational efficiency of the algorithm.

Assuming a homogeneous distribution of points, Equation 4.2 derives how the fraction of points outside the cylinder grows for a unit cylinder with $r = 1$.

$$
\begin{aligned}
f_{\text{outside}} &= \frac{V_{\text{outside}}}{V_{\text{total}}} = \frac{V_{\text{sphere}} - V_{\text{cyl}}}{V_{\text{sphere}}} \\
&= 1 - \frac{V_{\text{cyl}}}{V_{\text{sphere}}} \\
&= 1 - \frac{\pi s}{\frac{4}{3}\pi R^3} \\
&= 1 - \frac{3s}{4R^3}
\end{aligned}
\tag{4.2}
$$

By substituting the step size s using the geometric constraint derived in Equation 4.1 for a unit cylinder ($s = 2\sqrt{R^2 - 1}$), we can express this fraction entirely as a function of the query radius R:

$$
\begin{aligned}
f_{\text{outside}} &= 1 - \frac{3\left(2\sqrt{R^2 - 1}\right)}{4R^3} \\
&= 1 - \frac{3\sqrt{R^2 - 1}}{2R^3}
\end{aligned}
\tag{4.3}
$$

While Equation 4.3 models the radial waste outside the cylinder, it treats each query independently. In practice, consecutive spherical queries along the LoS overlap. Figure 4.2 visualizes how the fractions of the relevant and irrelevant queried volumes evolve as a function of the normalized query radius R, including the effect of this overlap.

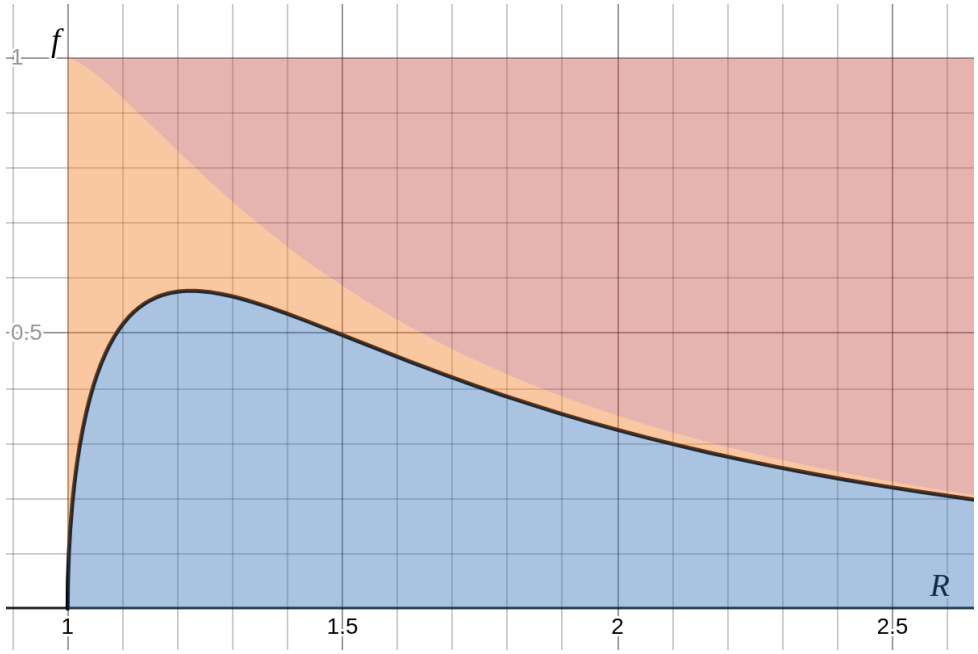


Figure 4.2: The evolution of query fractions f for different query radii R. Relevant volume is represented in blue, overlapping volume in orange, and radial waste in red.

As visualized in the figure, the fraction of radial waste (f_{outside}) approaches zero as R shrinks toward 1. However, in this state, the overlapping volume between consecutive spheres approaches 100%. Conversely, as R grows, the cubic expansion of the sphere causes the radial waste to approach 1.

Segmenting vegetation

In order to produce accurate viewsheds, vegetation segmentation is of critical importance. Because the algorithm treats vegetation as semi-transparent rather than fully opaque, confusing vegetation with solid obstacles will compromise the reliability of the results. This section verifies that the Myria3D deep learning model introduced in Section 2.3.2 is robust enough for downstream mission planning applications, and gives recommendations regarding the classification threshold.

To evaluate the Myria3D model introduced in Section 2.3.2, we use the partial False Detection Rate (FDR) as a performance metric, as a direct ground truth dataset of vegetation is unavailable. FDR measures the percentage of positive classifications that are false based on collisions with other human-verified classes in the referenced dataset. While not a perfect measure of absolute accuracy, FDR provides a reliable way to compare different methods. Figures 4.3–4.6 plot different methods using FDR and total detection rate.

A simple and widely used technique for segmenting LiDAR point clouds is a rule-based classifier that relies on some set of attributes and thresholds to segment a specific class like vegetation (Rapinel & Hubert-Moy, 2021). Commonly used LiDAR attributes are the number of returns, local planarity, height above ground, and the NDVI when available. For comparison, we first construct a theoretically optimal threshold using these features. This way we get an upper limit for what an optimal threshold-based ruleset could achieve for any given true vegetation coverage percentage.

Figure 4.3 shows the full sweep plotted over all features. As total detection rate passes 30%, FDR increases dramatically.

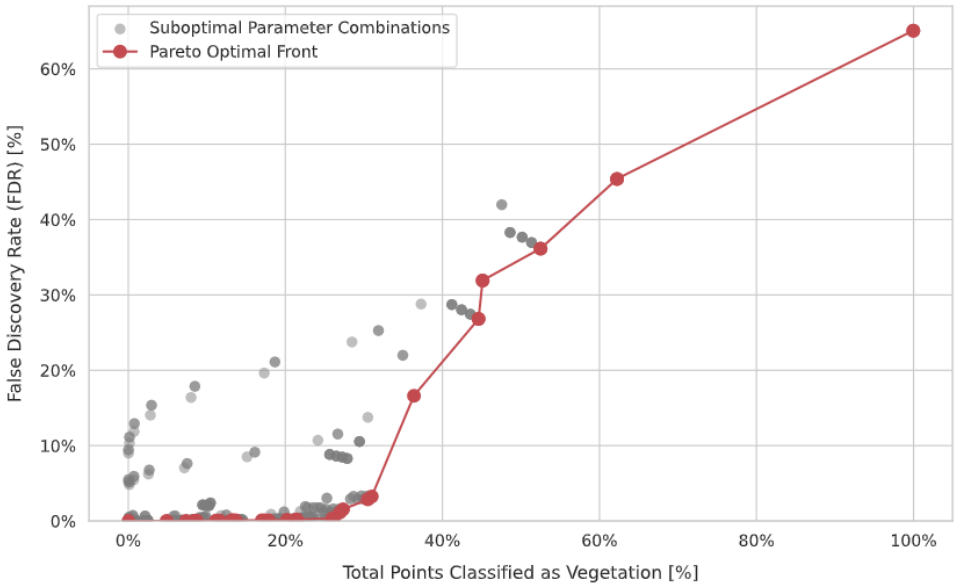


Figure 4.3: Pareto optimal front for the theoretical rule-based vegetation classifier.

Figure 4.4 shows the value of each feature on the rule-based classifier. It shows that the theoretically optimal rule-based classifier is very sensitive to specific thresholds and depends heavily on the unknown true vegetation percentage.

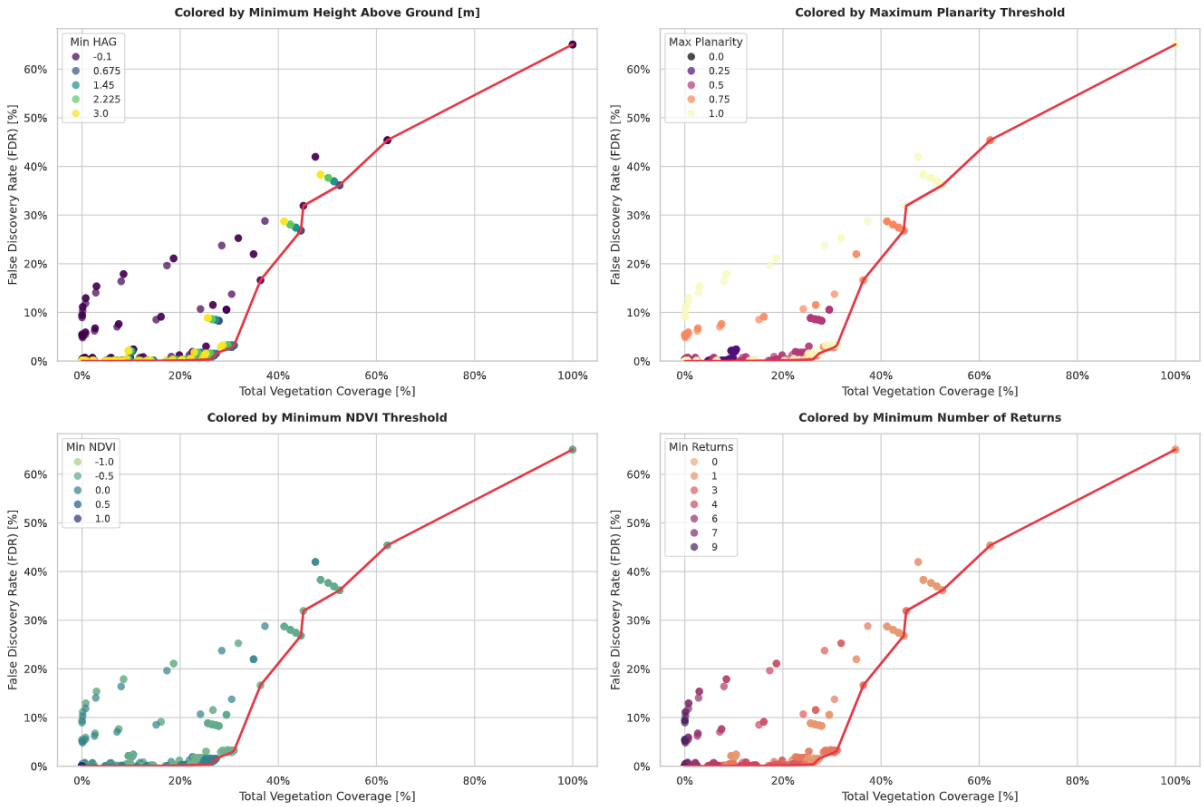


Figure 4.4: Diagnostic plots showing the impact of individual feature thresholds on the performance of the theoretical rule-based classifier.

Figure 4.5 compares the theoretically optimal rule-based to the Myria3D model and shows that all Myria3D variants completely dominate the optimal rule-based method. While the Myria3D method is slower, it does guard against extreme errors by only covering part of the total detection rate spectrum even for extreme values. The rule-based approach offers no such protections and is unlikely to generalize to other scenes.

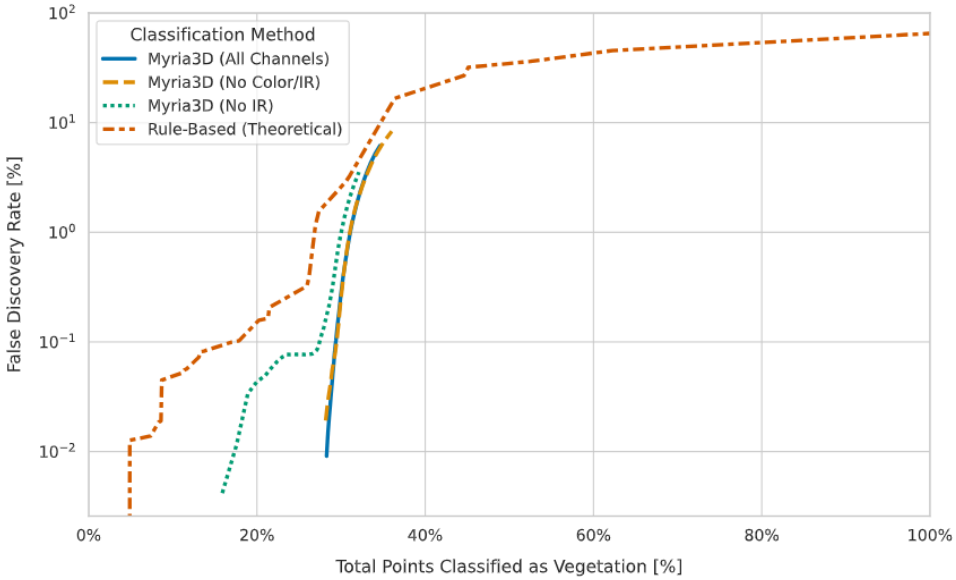


Figure 4.5: Comparison of the different classifiers.

Figure 4.6 shows the performance of the different Myria3d variants in detail. While the models with all channels and no RGBI perform similarly on this test, the model without the NIR channels is significantly worse. This could indicate an over-reliance on the NIR channel.

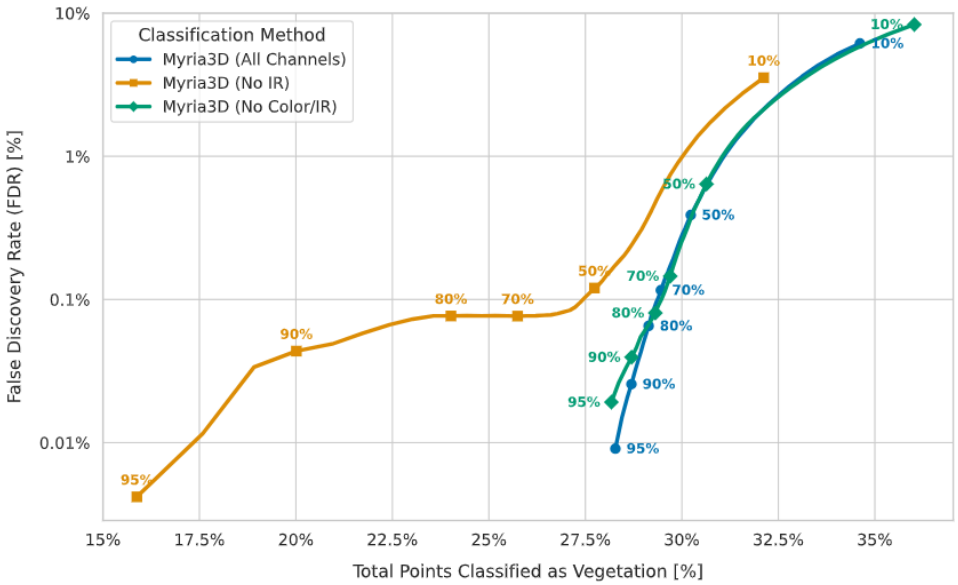


Figure 4.6: Comparison of the different input channels to the Myria3D classifier.

Vegetation modeling

An interesting question one might ask is "how much does vegetation affect visibility?". However, assessing the impact of vegetation on an "average" scene is not a useful goal, as uniform scenes do not exist. First, the ratio of vegetation to buildings and ground is highly variable. Second, results are extremely sensitive to the spatial arrangement of obstacles and the source point's position, a sensitivity that is amplified in 3D analysis. Finally, correcting for class ratios is difficult because a class's point count is not necessarily proportional to its volume. One example of this is how building facades are sparse due to the sensor angle, whereas vegetation is dense because of multiple internal laser bounces. Even though we can not easily answer the question for an average scene, including vegetation remains an intuitive improvement. We even show that when vegetation is excluded, and our 3D method is applied to buildings and ground, it is still more accurate than current 2.5D raster-based methods.

As discussed in Section 2.4.3, compensating for seasonal variations (summer-winter differences) can be done by calibrating the attenuation threshold using repeat data collection. When we extend this reasoning to operational scenarios, a large winter dataset could be preprocessed and corrected without requiring full overlap. Calibrating the attenuation coefficient (k) could likely be done using a relatively small, recent survey, like a localized UAV flight. This allows a large out-of-season dataset to be corrected efficiently using minimal in-season data, which is likely harder to collect.

Determining specific tree characteristics can further refine our model of visibility. A fundamental distinction between coniferous and deciduous trees could be established using biome maps or altitude data. Alternatively, change detection between summer and winter satellite or LiDAR imagery could identify deciduous trees through seasonal leaf loss. Automatic tree classification directly from LiDAR or optical imagery is another option, potentially avoiding the need for extra data collection. While classifying individual tree species is becoming increasingly feasible, it likely provides diminishing returns compared to the

basic deciduous-coniferous distinction. Therefore, species-level classification should not be prioritized over more impactful improvements.

Roof segmentation and clustering

Before the facade enhancement discussed in Section 2.3.2, we first segment and cluster roof surfaces using a simple rule-based approach. We filter points based on two criteria: the `building` class label and their surface normal. This results in a `roof` class marked in red in Figure 4.7.



Figure 4.7: Point cloud with segmented roofs highlighted in red.

Then, the roof points are clustered using a 2D DBSCAN algorithm, which relies on defined eps and min_points thresholds to group distinct structures. An example of the final classes is presented in Figure 4.8.



Figure 4.8: Roof points clustered into distinct building footprints using the DBSCAN algorithm, represented by alternating colors.

Finally, we construct concave hulls for each cluster, controlled by specific concavity thresholds. While all of these steps and parameters can introduce minor imperfections, the method performs adequately and is not a primary source of overall model error.

Regardless, future work could rigorously tune these parameters similarly to the method used in Section 4.3.2. This could be done by using datasets like the 3DBAG as ground truth for training and testing. To increase performance further, one could consider incorporating additional thresholds to better separate overlapping structures, or transitioning to more advanced classification strategies similar to those used in Section 2.3.2.

Temporal dynamics

An exciting extension of direct point cloud processing methods would be the integration of a temporal dimension. Here we discuss one possible implementation of such a 4D point cloud. The core concept assigns a "momentum" value to each point based on how often or how quickly it changes over time. Examples of this idea are already present in Chapter 2. For example, trees have moderate to high momentum. Not only do they gain and lose leaves with the seasons, but they also grow each year. Buildings would have low momentum, being mostly constant for several years while some are built or demolished. Ground points would have very low momentum, only changing through natural processes or large infrastructure projects.

This temporal dimension could be highly relevant for tactical planning. For example, cars will have a very high momentum, signaling they provide very unreliable cover for infiltrators. The time gap between data capture and mission execution is often long enough for vehicles to move. In some cases, they may even be moving during the capture itself. This new metric could enable better planning by giving users control over what points to include as obstacles given their momentum and the recency of the data.

On the other hand, this addition would reveal unclassified points with low momentum that maintain a consistent footprint across multiple surveys. This could give users the confidence to include them as permanent obstacles, even if their semantics are unknown. Additionally, this momentum could serve as a feature for further improving semantic segmentation itself, distinguishing dynamic objects like cars from static vegetation. The AHN could be a suitable dataset to explore these 4D methods, as it provides multiple overlapping historical versions.

4.2 Answers to research questions

While already discussed narratively in Section 2.6, this section explicitly answers the research questions laid out in Chapter 1.

1. *How can point clouds be automatically and efficiently accessed for performing LoS visibility queries using available frameworks?*

The proposed method demonstrated how large remote ALS datasets can be efficiently accessed and queried by utilizing the Cloud Optimized Point Cloud (COPC) format for streaming, combined with optimized spatial indexing for rapid localized fetching.

2. *How can vegetation be automatically classified in point clouds using known methods?*

This thesis shows that vegetation can be automatically and reliably classified using deep learning techniques. Specifically, the Myria3D model generalizes well to the AHN dataset and dominates theoretically optimal rule-based classifiers.

3. *How can permeability parameters be derived from point-cloud features so that they fit within an exponential attenuation model of visibility?*

This thesis adapts the Beer-Lambert law to more realistically model semi-transparent vegetation. Permeability parameters are derived by using local point density to calculate an exponential attenuation coefficient directly along the line of sight.

4. *How can 3D LoS calculation best be visualized for use in military airborne mission planning?*

The resulting 3D visibility analyses can be effectively visualized using either 3D tools like GIS software and web viewers, or by extracting 2D map overlays and sections for conventional planning tools.

5. *How does incorporating semi-transparent vegetation influence the accuracy and performance of visibility calculations?*

Incorporating semi-transparent vegetation significantly improves accuracy compared to binary models (which underestimate visibility) and opaque models (which overestimate it). Furthermore, it gives commanders the flexibility to implement appropriate risk management, ultimately increasing the area in which missions can effectively operate.

6. *What is the accuracy of the proposed method compared to existing methods and compared to reality?*

The new method outperforms industry-standard raster techniques given the same input data and output resolution. Additionally, synthetically densifying building facades mitigates inherent scanning sparsity. This improves overall accuracy and justifies the potential introduction of minor geometric errors. It allows us to increase the spatial resolution of our results by enabling thinner LoS cylinders without introducing significant errors.

> *How can military aerial mission planning be improved by incorporating semi-transparent vegetation in probability-based visibility calculations computed directly on airborne LiDAR point clouds?*

Ultimately, military airborne mission planning can be substantially improved by switching to a non-binary, fully 3D model that mathematically translates physical point density into visual occlusion. Our methodology provides a viable alternative to current workflows and offers distinct benefits in terms of structural accuracy and analytical flexibility. These new workflows enable analysts to make context-aware decisions, making military mission planning more effective across a wide variety of tactical scenarios.

4.3 Main contributions

The main contribution of this thesis is improving fundamental visibility analysis methods, particularly using direct point cloud analysis and a non-binary visibility model.

By applying these improvements to airborne mission planning, this work contributes to more effective and safer military operations. As data gathering and processing become increasingly critical in modern conflicts, staying at the forefront of data science and AI is a matter of national security.

On a higher level, this research also supports the overarching goal of the Dutch Ministry of Defence and the Data Science Centre of Excellence in particular towards increasingly data-driven armed forces.

This research advances the concept of shifting analytical control downstream to the end-user. In traditional methods, critical parameters such as spatial resolution or a model's errors and risks are often hardcoded by the developer. A voxel or raster cell of space is simply either full or empty. The proposed methodology shifts some of this control to analysts, mission planners, and commanders, giving them the flexibility to tune visibility thresholds based on acceptable risk and mission goals.

From a scientific perspective, this thesis advances the "smart point cloud" paradigm. It demonstrates the practical value of moving away from simplified 2.5D, binary world models toward fully 3D, semantically rich, and probabilistic representations of the environment. By reinforcing that complex spatial analysis can be performed efficiently directly on raw LiDAR data, this research helps bridge some of the gap between low-level data and high-level spatial decision-making.

4.4 Design process

At the start of the thesis, I decided to adopt a modular architecture, meaning the different processing steps (query, enhance, calculate, visualize) were decoupled so I could easily swap any individual module for a newer version without having to modify the rest of the pipeline.

I am happy to say this worked very well. In past projects I have run into many issues that were avoided this time. Because of the architecture, I could first focus on a basic implementation of each module before starting to iterate. This meant I could generate basic results quite early in the process.

For querying the data, I was able to build up the implementation in iterations. First, I used a local test file, then a local download of the AHN, then remote fetching of AHN6, then remote fetching of all AHN versions including NIR, then adding LidarHD, and then adding CanElevation.

Once I had all this functionality implemented, I decided to release the query module as a separate tool. Because the query module was completely decoupled from the visibility calculations, I was able to spin off this data-fetching logic relatively easily into a standalone Python package. The package is called Cloudfetch and is available as free and open-source software on GitHub[38] and through PyPI[39].

Cloudfetch allows users to easily download arbitrary polygons of large remote point cloud datasets in Python using optimized COPC queries under the hood. This package was only useful for my research, but can be applied to many other problems and projects in the future. It also allows users to easily add their own proprietary or classified data sources, ensuring the tool remains useful for military contexts where public data is insufficient.

The modular design also made it possible to iterate on the enhancement and pre-processing module. This also allowed me to work on other modules in parallel when I was stuck or waiting for feedback.

First, I implemented a simple rule-based classifier to isolate vegetation. Once the pipeline was stable, I returned to this module to integrate the more advanced, deep learning approach used in the final pipeline. I was able to do this without disrupting the downstream calculations, and I could use the rule-based module as a baseline going forward. Similarly, the synthetic facade enhancement algorithm was developed and injected into the pipeline only when the need for it became clear to me.

For the calculating module, I slowly built up the capabilities over the course of the project:

1. the simplest point-to-point calculation, which computes the visibility between two 3D points
2. point to multiple points, which computes the visibility between a source point and multiple target points.
3. point to area, which computes the viewshed from a source point to a surrounding area.
4. point to volume, which computes the viewshed from a source point to a fully 3D volume.
5. multiple points to volume, which computes the cumulative viewsheds from multiple observers to a fully 3D volume. Each successive step allowed for new applications that I could immediately start implementing and testing as the pipeline was always working.

Finally, the modular architecture made it so I could start iterating and improving my visualization module as soon as the first implementation of the pipeline was running. Additionally, it made it easy to spin off part of the code that implements the webviewer[40]. This made it so I could push the webviewer to the server, available at the courtesy of the GDMC group at TU Delft, without them having to host my entire codebase.

4.5 Goals

A MoSCoW analysis is included in Table 5. It sorts all the different goals set out at the start of the thesis into categories and orders them by priority. In this hierarchy: “Must” is needed to reach the minimum viable product; “Should” constitutes a successful product; “Could” contains goals that go beyond what is needed for success but are nice to have if time allows; and “Won’t” explicitly states goals that are out of scope. In Table 5, colors indicate the status of the initial goals: green for achieved, and red for unachieved.

[38] https://github.com/yairroorda/cloudfetch
[39] https://pypi.org/project/cloudfetch/
[40] https://gist.bk.tudelft.nl/apps/PointCloudLineOfSight/

	Prepare data	Calculate visibility	Present results
Must	Develop a pipeline that works for the AHN dataset Segment vegetation in AHN using a rule-based system	Come up with or extend a fully 3D method to compute probable visibility, taking into account vegetation directly on point clouds	Produce probable visibility viewshed maps of multiple representative or interesting subsets of the point cloud data Deliver a thesis document, 3 presentations, and a code repository
Should	Make it possible to process any arbitrary tile of the AHN dataset	Apply this method to the case of military helicopter planning.	Visualize resulting visibility queries in 3D maps Make an extra effort to share results with DCSE, like through a presentation/summary/ poster
Could	Segment trees using a deep-learning based method Include textured AHN6 data Add species information Modify density based on seasonal changes	Optimize the method for scaling and performance (maybe LoD?)	Prepare a paper to submit to a conference/journal Make an online demo or plugin
Won't	Data fusion approaches with satellite imagery, photogrammetry, or radar Real-time or onboard processing of data	Investigating sound or other sources of non-binary visibility like glare, weather, or data accuracy	Apply to other problems outside of airborne mission planning Perform field validations or capture additional data.

Table 5: MoSCoW analysis with colors indicating what goals were achieved.

Table 5 shows that while the wording was not as precise at the start of the project as it could be now, most goals were achieved. For data preparation, this means all 'Must' and 'Should' requirements were successfully implemented. Users can not only process any arbitrary part of the AHN dataset but also of the LidarHD and CanElevation datasets. Extra time was spent on implementing and evaluating the Myria3D model. Species information of different trees was the only thing not implemented due to time constraints.

For the visibility calculations, this means a repeatable method was implemented, verified, and applied to multiple problems relevant to mission planning. Presenting the results was achieved through this document, a presentation at the DSCE conference, the publication of

the Cloudfetch PyPI package, the release of an online viewer, and the application for the NCG symposium 2026.

As with any scientific research, the main objective of this work is to contribute new knowledge. In the case of a thesis, it must also prove the needed skills to receive the MSc in Geomatics degree.

This thesis covers many of the core learning objectives for the Geomatics program and covers the full breadth of geoinformation processing. This starts with understanding the physics of LiDAR remote sensing measurements, the different attributes, and how they can be used effectively. Next, efficiently storing, retrieving, and processing geospatial data, in this case LiDAR, by using optimized indexing and streaming methods. Writing software to process and analyse (3D) data, adding semantics, refining and verifying. Finally, it covers visualization of spatial analysis both through GIS tools and a custom web viewer.

Bibliography

Alblas, L. (2012). Archaeological visibility analysis with GIS. *The Council of Europen Geodetic Surveyors*, 1–7. https://www.clge.eu/wp-content/uploads/2012/05/CLGE-SC2012_Alblas.-THE-NETHERLANDS__Archaeological-visibility-analysis-with-GIS.pdf

Alsadik, B., Gerke, M., & Vosselman, G. (2014). Visibility analysis of point cloud in close range photogrammetry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 9–16. https://doi.org/10.5194/isprsannals-II-5-9-2014

Bartie, P., Reitsma, F., Kingham, S., & Mills, S. (2010). Advancing visibility modeling algorithms for urban environments. *Computers, Environment and Urban Systems*, *34*(6), 518–531. https://doi.org/10.1016/j.compenvurbsys.2010.06.002

Bartie, P., Reitsma, F., Kingham, S., & Mills, S. (2011). Incorporating vegetation into visual exposure modeling in urban environments. *International Journal of Geographical Information Science*, *25*, 851–868. https://doi.org/10.1080/13658816.2010.512273

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, *18*(9), 509–517. https://doi.org/10.1145/361002.361007

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., & Taubin, G. (1999). The Ball-Pivoting Algorithm for Surface Reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, *5*, 349–359. https://doi.org/10.1109/2945.817351

Chmielewski, S. (2021). Towards Managing Visual Pollution: A 3D Isovist and Voxel Approach to Advertisement Billboard Visual Impact Assessment. *ISPRS International Journal of Geo-Information*, *10*(10), 656. https://doi.org/10.3390/ijgi10100656

Dean, D. J. (1997). Improving the accuracy of forest viewsheds using triangulated networks and the visual permeability method. *Canadian Journal of Forest Research*, *27*(7), 969–977. https://doi.org/10.1139/x97-062

Dienst Geografie. (2024, October). *(Satelliet)beeldvoorziening en toepassingen Defensie*. https://www.ahn.nl/congres-ahn-beeldmateriaal-2024

Díaz-Vilariño, L., González-deSantos, L., Verbree, E., Michailidou, G., & Zlatanova, S. (2018). From Point Clouds to 3D Isovists in Indoor Environments. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 149–154. https://doi.org/10.5194/isprs-archives-XLII-4-149-2018

Elseberg, J., Magnenat, S., Siegwart, R. Y., & Nuechter, A. (2012). *Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration*. https://api.semanticscholar.org/CorpusID:8774377

European Commission. (2025). *White Paper for European defence - Readiness 2030* [Whitepaper]. https://commission.europa.eu/document/download/e6d5db69-e0ab-4bec-9dc0-3867b4373019_en?filename=White%20paper%20for%20European%20defence%20%E2%80%93%20Readiness%202030.pdf

Fisher, P. (1994). *Probable and fuzzy models of the viewshed operation*. CRC Press.

Fisher-Gewirtzman, D., Shashkov, A., & Doytsher, Y. (2013). Voxel based volumetric visibility analysis of urban environments. *Survey Review*, *45*(333), 451–461. https://doi.org/10.1179/1752270613Y.0000000059

Gaydon, C. (2022). *Myria3D: Deep Learning for the Semantic Segmentation of Aerial Lidar Point Clouds*. https://github.com/IGNF/myria3d

Gaydon, C., Daab, M., & Roche, F. (2024). *FRACTAL: An Ultra-Large-Scale Aerial Lidar Dataset for 3D Semantic Segmentation of Diverse Landscapes*. https://arxiv.org/abs/TBD

Gettinger, D. (2020). *The Drone Databook*. Center for the Study of the Drone at Bard College. https://dronecenter.bard.edu/files/2019/10/CSD-Drone-Databook-Web.pdf#page=26.06

Gheorghiu, I. (2024). *Analysis of the visibility of GPS satellites in the urban environment using point cloud representation* [Master's thesis]. https://www.gdmc.nl/publications/2024/MScThesis IrinaGheorghiu.pdf#page=67.01

Govindaraju, V., Leng, G., & Qian, Z. (2014). Visibility-based UAV path planning for surveillance in cluttered environments. *2014 IEEE International Symposium on Safety, Security, And Rescue Robotics (2014)*, 1–6. https://doi.org/10.1109/SSRR.2014.7017660

Granelli, F., Lu, Y.-P., Wu, Q., Yuan, Z., Abdalla, A. S., Marojevic, V., Jiang, Y., Afghah, F., Geraci, G., Mukherjee, A., Baidya, S., Grieco, G., Yaqoob, A., Hassan, M. A., Namuduri, K., Prasad, R. R. V., Lahlou, L., Sengendo, J., & Muntean, G.-M. (2026). A Survey on Unmanned Aerial Vehicles (UAVs) Communications: State-of-the-Art, Existing Standards, and Future Directions. *IEEE Open Journal of the Communications Society*, *7*, 3000–3045. https://doi.org/10.1109/OJCOMS.2026.3675046

Gresham, A., Healey, J. R., Eichhorn, M. P., Barton, O., Smith, A. R., & Shannon, G. (2023). Horizontal viewsheds of large herbivores as a function of woodland structure. *Ecology and Evolution*, *13*(11), e10699. https://doi.org/10.1002/ece3.10699

Guth, P. (2009). *Incorporating vegetation in viewshed and line-of-sight algorithms*. https://www.academia.edu/122367723/Incorporating_Vegetation_in_Viewshed_and_Line_Of_Sight_Algorithms

Guth, P. L. (2012). *Probabilistic line-of-sight with LiDAR point clouds*. https://www.asprs.org/a/publications/proceedings/Sacramento2012/files/Guth.pdf

het Waterschapshuis. (n.d.). *Downloads Actueel Hoogtebestand Nederland en Beeldmateriaal — basisdata.nl*.

Hook, D., Porter, S., & Herzog, C. (2018). Dimensions: Building Context for Search and Evaluation. *Frontiers in Research Metrics and Analytics*, *3*. https://doi.org/10.3389/frma.2018.00023

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2019). RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *Corr*. http://arxiv.org/abs/1911.11236

industryarc. (2025, January). *LiDAR for Security Market Research Report: Market size, Industry outlook, Market Forecast, Demand Analysis, Market Share, Market Report 2024-2030 — industryarc.com*.

Inglis, N. C., Vukomanovic, J., Costanza, J., & Singh, K. K. (2022). From viewsheds to viewscapes: Trends in landscape visibility and visual quality research. *Landscape and Urban Planning*, *224*, 104424. https://doi.org/10.1016/j.landurbplan.2022.104424

Kashyap, R., Bhuvan, M. S., Chamarti, S., Bhat, P., Jothish, M., & Annappa, K. (2014). Algorithmic Approach for Strategic Cell Tower Placement. *Modeling and Simulation 2014 5th International Conference on Intelligent Systems*, 619–624. https://doi.org/10.1109/ISMS.2014.112

Katz, S., & Tal, A. (2025). HPRO: Direct Visibility of Point Clouds for Optimization. *Computer Graphics Forum*, *44*. https://doi.org/10.1111/cgf.70046

Katz, S., Tal, A., & Basri, R. (2007). Direct visibility of point sets. *ACM Transactions on Graphics*, *26*(3), 24. https://doi.org/10.1145/1276377.1276407

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 61–70.

Leenders, N., Oijen, J. van, Lindelauf, R., & Cule, B. (2025). Uncertainty in visibility: a scoping review of the probable and fuzzy viewshed for observer location optimization. *International Journal of Geographical Information Science*, *0*, 1–30. https://doi.org/10.1080/13658816.2025.2581833

Llobera, M. (2007). Modeling visibility through vegetation. *International Journal of Geographical Information Science*, *21*, 799–810. https://doi.org/10.1080/13658810601169865

Lu, Y.-H., & Han, J.-Y. (2020). GNSS Satellite Visibility Analysis Based on 3D Spatial Information in Urban Areas. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 123–128. https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-123-2020

Maaiveld, T., Domela Nieuwenhuis, D., Reus, N. de, Schadd, M., & Kuijper, F. (2024). *Where to go and how to get there: Tactical terrain analysis for military unmanned ground-vehicle mission planning*. https://doi.org/10.1007/978-3-031-71397-2_7

Meagher, D. (1982). Geometric Modeling Using Octree-Encoding. *Computer Graphics and Image Processing*, *19*, 129–147. https://doi.org/10.1016/0146-664X(82)90104-6

Mehra, R., Tripathi, P., Sheffer, A., & Mitra, N. J. (2010). Visibility of noisy point cloud data. *Computers & Graphics*, *34*(3), 219–230. https://doi.org/10.1016/j.cag.2010.03.002

Ministerie van Defensie. (2014). *Nederlandse Doctrine voor Air & Space Operations*. 30. https://www.defensie.nl/downloads/publicaties/2014/12/18/nederlandse-doctrine-voor-air%E2%80%93space-operations

Ministerie van Defensie. (2023). *Defensie Strategie Data Science en AI 2023-2027* [Technical report]. Ministerie van Algemene Zaken. https://www.rijksoverheid.nl/documenten/rapporten/2023/05/31/defensie-strategie-data-science-en-artificiele-intelligentie-2023-2027

Ministerie van Defensie. (2024). *Defence White Paper* [Whitepaper]. Ministerie van Defensie. https://english.defensie.nl/downloads/publications/2024/09/05/defence-white-paper-2024

Mistick, K. A., Campbell, M. J., Thompson, M. P., & Dennison, P. E. (2023). Using airborne LiDAR and machine learning to predict visibility across diverse vegetation and terrain

conditions. *International Journal of Geographical Information Science*, *37*(8), 1728–1764. https://doi.org/10.1080/13658816.2023.2224421

Mistick, K. A., Dennison, P. E., Campbell, M. J., & Thompson, M. P. (2022). Using Geographic Information to Analyze Wildland Firefighter Situational Awareness: Impacts of Spatial Resolution on Visibility Assessment. *Fire*, *5*(5), 151. https://doi.org/10.3390/fire5050151

Murgoitio, J. J., Shrestha, R., Glenn, N. F., & Spaete, L. P. (2013). Improved visibility calculations with tree trunk obstruction modeling from aerial LiDAR. *International Journal of Geographical Information Science*, *27*(10), 1865–1883. https://doi.org/10.1080/13658816.2013.767460

Natijne, A. van. (n.d.). *GeoTiles: readymade geodata with a focus on the Netherlands — geotiles.citg.tudelft.nl*.

Nebiker, S., Bleisch, S., & Christen, M. (2010). Rich point clouds in virtual globes - A new paradigm in city modeling?. *Computers, Environment and Urban Systems*, *34*(6), 508–517. https://doi.org/10.1016/j.compenvurbsys.2010.05.002

Orlof, J., Ozimek, P., Łabędź, P., Widłak, A., & Ozimek, A. (2024). Generating viewsheds based on the digital surface model (DSM) and point cloud. *PLOS ONE*, *19*(12), e312146. https://doi.org/10.1371/journal.pone.0312146

Oude Elberink, S. (2010). *Acquisition of 3D topography: automated 3D road and building reconstruction using airborne laser scanner data and topographic maps* [PhD Thesis]. https://doi.org/10.3990/1.9789061642879

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *The BMJ*, *372*, n71. https://doi.org/10.1136/bmj.n71

PDAL Contributors. (2025, August). *PDAL Point Data Abstraction Library*. https://doi.org/10.5281/zenodo.10884408

Peters, R., Ledoux, H., & Biljecki, F. (2015). *Visibility Analysis in a Point Cloud Based on the Medial Axis Transform*. https://doi.org/10.2312/UDMV.20151342

Poux, F., Hallot, P., Neuville, R., & Billen, R. (2016). Smart point cloud: definition and remaining challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 119–127. https://doi.org/10.5194/isprs-annals-IV-2-W1-119-2016

Pyysalo, U., Oksanen, J., & Sarjakoski, T. (2009). *Viewshed analysis and visualization of landscape voxel models*. https://api.semanticscholar.org/CorpusID:17327676

Rapinel, S., & Hubert-Moy, L. (2021). One-Class Classification of Natural Vegetation Using Remote Sensing: A Review. *Remote Sensing*, *13*(10). https://doi.org/10.3390/rs13101892

Rasova, A. B. (2018). Vegetation modelling in 2.5D visibility analysis. *Cartographic Letters*, *26*(1), 10–20. https://www.researchgate.net/publication/330957864_Vegetation_modelling_in_25D_visibility_analysis

Reddy, R., & Sinha, S. (2025). State-of-the-Art Review: Electronic Warfare Against Radar Systems. *IEEE Access*, *13*, 57530–57567. https://doi.org/10.1109/ACCESS.2025.3555493

Ruzickova, K., Ruzicka, J., & Bitta, J. (2021). A new GIS-compatible methodology for visibility analysis in digital surface models of earth sites. *Geoscience Frontiers*, *12*(4), 101109. https://doi.org/https://doi.org/10.1016/j.gsf.2020.11.006

Samet, H. (1984). The Quadtree and Related Hierarchical Data Structures. *ACM Comput. Surv.*, *16*(2), 187–260. https://doi.org/10.1145/356924.356930

Sharma, R. (2026, May). *LiDAR for Defense Market Research Report 2034 — marketintelo.com*. https://marketintelo.com/report/lidar-for-defense-market

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wang, J.-H., Tian, Y.-Y., Chen, B., & Wang, P.-S. (2025). Neural Visibility of Point Sets. *Arxiv.org*. https://doi.org/10.1145/3757377.3763869

Wheatley, D. (1995). Cumulative viewshed analysis: a GIS-based method for investigating intervisibility, and its archaeological application. *Archaeology and Geographic Information Systems: A European Perspective*. https://www.academia.edu/705833/Cumulative_viewshed_analysis_a_GIS_based_method_for_investigating_intervisibility_and_its_archaeological_application

Yaagoubi, R., Yarmani, M. E., Kamel, A., & Khemiri, W. (2015). HybVOR: A Voronoi-Based 3D GIS Approach for Camera Surveillance Network Placement. *ISPRS International Journal of Geo-Information*, *4*(2), 754–782. https://doi.org/10.3390/ijgi4020754

Zhang, G., Yang, D., & Cheng, S. (2025). Voxelized Point Cloud and Solid 3D Model Integration to Assess Visual Exposure in Yueya Lake Park, Nanjing. *Land*, *14*(10). https://doi.org/10.3390/land14102095

Zhang, G.-T., Verbree, E., & Wang, X.-J. (2021). An Approach to Map Visibility in the Built Environment From Airborne LiDAR Point Clouds. *IEEE Access*, *9*, 44150–44161. https://doi.org/10.1109/ACCESS.2021.3066649

Zhao, J., Xu, A., Zhang, X., Zhang, Y., Zhang, Y., Cao, M., & Huang, M. (2024). Overview and Prospects of Visibility Analysis Approaches. *Proceedings*, *110*(1), 28. https://doi.org/10.3390/proceedings2024110028

Zong, X., Wang, T., Skidmore, A. K., & Heurich, M. (2021). The impact of voxel size, forest type, and understory cover on visibility estimation in forests using terrestrial laser scanning. *Giscience & Remote Sensing*, *58*(3), 323–339. https://doi.org/10.1080/15481603.2021.1873588

Zong, X., Wang, T., Skidmore, A. K., & Heurich, M. (2023). LiDAR reveals a preference for intermediate visibility by a forest-dwelling ungulate species. *Journal of Animal Ecology*, *92*(7), 1306–1319. https://doi.org/10.1111/1365-2656.13847

A Reproducibility self-assessment

This self-assessment evaluates the reproducibility of the thesis based on "the four R's" framework proposed by Inglis et al. (2022) shown in Figure A.1. All data and nearly all code related to this thesis are openly accessible online through the referenced GitHub repositories and the PyPI package registry. Complete instructions and package management environments are provided in the project repository to facilitate seamless reconstruction of the workflow. Therefore, I rate the reproducibility of this thesis as high.

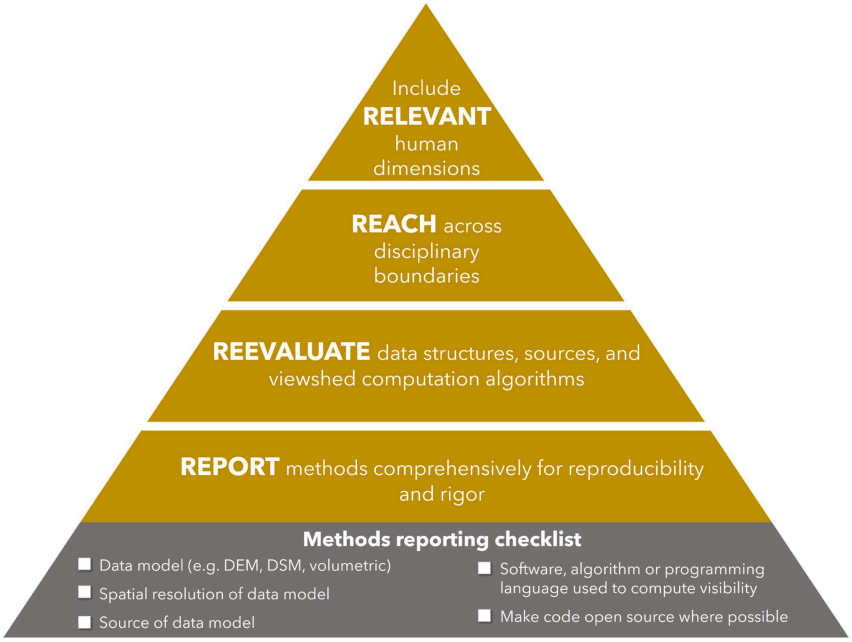


Figure A.1: The Four R's: A suggested framework for reporting and planning future visibility research to efficiently build on previous work, encourage interdisciplinary applications and support reproducibility and rigor (Inglis et al., 2022)

This thesis only uses open data, no classified data sources were used. The LiDAR datasets used are the Dutch AHN, French LidarHD, and Canadian CanElevation. These all carry open licenses, making them suitable for reuse and research. The custom `Cloudfetch` data querying module has been published open-source via PyPI and GitHub.

Furthermore, the complete codebase is open-source and available on GitHub[41] with the exception of the optimized route planning algorithm due to ethical concerns. This includes the core method as well as the scripts used for evaluation. To reproduce setups across machines and operating systems, the repository also includes `pixi.toml` and `pixi.lock` environment configuration files.

This research relies exclusively on open-source tools and software. This includes the CloudCompare point cloud viewer[42], the QGIS[43] long term release 3.34, and Potree framework[44].

[41] https://github.com/yairroorda/Thesis
[42] https://www.cloudcompare.org/
[43] https://www.qgis.org/
[44] https://github.com/potree/potree

B Declaration of LLM usage

This document benefited from the use of LLM tools. Grammarly[45] was used for grammar and spelling suggestions. Connected Papers[46] was used as part of the literature discovery workflow. Additionally, the Perplexity[47] and Gemini[48] models were used in combination with other sources when searching for background information and writing, especially when related to programming and debugging. The Gemini 3.1 Pro model was used to review the text and provide feedback. All results, analyses, and conclusions are my own, and I have verified that the content is original and meets academic integrity standards.

[45] https://www.grammarly.com/
[46] https://www.connectedpapers.com/
[47] https://www.perplexity.ai/
[48] https://gemini.google.com/app

C Keyword queries

Below are the full keyword queries used in Section 2.2 to search each of the three selected databases.

Scopus

```
(TITLE-ABS-KEY (visibility OR viewshed OR "view shed" OR "line of sight" OR
intervisibility OR "line-of-sight")

AND TITLE-ABS-KEY ("fuzzy" OR "probable viewshed" OR "probability of visibility"
OR "probabilistic visibility" OR "partial visibility" OR "visual permeability" OR
"cumulative visibility" OR "continuous visibility")

AND TITLE-ABS-KEY (LiDAR OR lidar OR "airborne laser scanner" OR "airborne laser
scanning" OR ALS OR "terrestrial laser scanner" OR TLS OR "laser scanning" OR
"point cloud*" OR "voxel grid" OR "digital surface model" OR DSM OR "digital
elevation model" OR DEM OR photogrammetry OR "close-range photogrammetry"))
```

The Lens

```
visibility OR viewshed OR "view shed" OR "line of sight" OR intervisibility OR
"line-of-sight"

AND ("fuzzy visibility" OR "fuzzy viewshed" OR "probable viewshed" OR "probability
of visibility" OR "probabilistic visibility" OR "partial visibility" OR "visual
permeability" OR "cumulative visibility" OR "continuous visibility")

AND (LiDAR OR lidar OR "airborne laser scanner" OR "airborne laser scanning" OR
ALS OR "terrestrial laser scanner" OR TLS OR "laser scanning" OR "point cloud*" OR
"voxel grid" OR "digital surface model" OR DSM OR "digital elevation model" OR DEM
OR photogrammetry OR "close-range photogrammetry")
```

Dimensions

```
visibility OR viewshed OR "view shed" OR "line of sight" OR intervisibility OR
"line-of-sight"

AND ("fuzzy visibility" OR "fuzzy viewshed" OR "probable viewshed" OR "probability
of visibility" OR "probabilistic visibility" OR "partial visibility" OR "visual
permeability" OR "cumulative visibility" OR "continuous visibility")

AND (LiDAR OR lidar OR "airborne laser scanner" OR "airborne laser scanning" OR
ALS OR "terrestrial laser scanner" OR TLS OR "laser scanning" OR "point cloud*" OR
"voxel grid" OR "digital surface model" OR DSM OR "digital elevation model" OR DEM
OR photogrammetry OR "close-range photogrammetry")
```