Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Enhancing GEDI Accuracy by Combining Different Geolocation Correction Criteria and Parallel Processing Methods

Leonel Luís da Silva Corado

Orientador(es) | Teresa Gonçalves

Sérgio Rui Godinho

Évora 2024

Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Enhancing GEDI Accuracy by Combining Different Geolocation Correction Criteria and Parallel Processing Methods

Leonel Luís da Silva Corado

Orientador(es) | Teresa Gonçalves

Sérgio Rui Godinho

Évora 2024

A dissertação foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor da Escola de Ciências e Tecnologia:

Presidente	|	Vasco Pedro (Universidade de Évora)
Vogais	|	Luís Rato (Universidade de Évora) (Arguente) Sérgio Rui Godinho (Universidade de Évora) (Orientador)

Évora 2024

O Green World
Don't desert me now
Made of you,
and you of me

Acknowledgements

First of all, I would like to express my gratitude to my advisors: Professor Teresa, for her creativity, compassionate mentorship, and genuine support in writing this thesis, and Dr. Sérgio, for his remarkable expertise and motivational presence in steering me on the right path. Both helped me grow academically and professionally, a big thank you.

This work was conducted within the framework of the GEDI4SMOS project (Combining LiDAR, radar, and multispectral data to characterize the three-dimensional structure of vegetation and produce land cover maps), financially supported by the Directorate-General for Territory (DGT) with funds from the Recovery and Resilience Plan (Investimento RE-C08-i02: Cadastro da Propriedade Rústica e Sistema de Monitorização da Ocupação do Solo), which I would like to thank. A special thank you to Professor Pedro Salgueiro for setting up the NIIAA cluster for the experiments conducted throughout this dissertation.

I would also want to thank all of my friends, who helped me grow throughout my years at the University of Évora, a thank you to the whole Insetos group, especially to Daniela for all the support, Helena for bringing joy to life, Joel & Abigail for believing in me, Mara for helping me grow, Tomás for all of the TCG and Rocket League gaming sessions, Teimas for being a true inspiration both culturally and professionally and to all of my work colleagues, and to everyone - thank you!

I am also deeply grateful to my Taekwondo Master, Joaquim Dias, for helping me channel my indomitable spirit and manage the stress that accompanied the writing of this thesis.

Finally, I want to thank my dad Leonel and my little brother Pedro for their relentless support throughout my life, for always believing in me, for the laughs and for being my refuge in the hardest of times.

Contents

List of Figures

List of Tables

Acronyms

3D Three-Dimensional

AGB Aboveground Biomass

AGBD Aboveground Biomass Density

ALS Airborne Laser Scanning

ATLAS Advanced Topographic Laser Altimeter System

CHM Canopy Height Model

CPU Central Processing Unit

CRS Coordinate Reference System

CRSSDA Curve Root Sum Squared Differential Area

CSV Comma-Separated Values

DEM Digital Elevation Model

EO Earth Observation

EPSG European Petroleum Survey Group

ER Electromagnetic Radiation

FHD Foliage Height Diversity

FWHM Full Width Half Maximum

GEDI Global Ecosystem Dynamics Investigation

GIS Geographic Information Systems

GPS Global Positioning System

GPU Graphics Processing Unit

GPKG Geopackage

HOMER High Output Maximum Efficiency Resonator

ICESat-2 Ice, Cloud and land Elevation Satellite 2

IMU Inertial Measurement Unit

ISS International Space Station

JEM-EF Japanese Experiment Module-Exposed Facility

KL Kullback-Leibler

LiDAR Light Detection and Ranging

MAE Mean Average Error

NASA LP DAAC NASA Land Processes Distributed Active Archive Center

PAI Plant Area Index

PAVD Plant Area Volume Density

RH Relative Heights

RMSE Root Mean Squared Error

R^2 Coefficient of Determination

RAM Random Access Memory

RS Remote Sensing

SAR Synthetic Aperture Radar

SG Setting Groups

SHP Shapefile

SRTM Shuttle Radar Topography Mission

WGS84 World Geodetic System 84

Abstract

As global environmental challenges intensify, monitoring terrestrial ecosystems has become crucial for addressing climate change. Spaceborne LiDAR missions, such as NASA's Global Ecosystem Dynamics Investigation (GEDI), play a key role in quantifying Earth's vegetation structure and land cover. GEDI provides high-resolution measurements of forest structure and topography, but these readings are often affected by geolocation errors caused by satellite platform instability and atmospheric interference, compromising the accuracy of canopy height and terrain elevation estimates. Existing geolocation correction methods, such as the GEDI Simulator, apply orbit-level corrections, which prove inadequate for heterogeneous landscapes.

This dissertation introduces *GEDICorrect*, a novel framework for footprint-level geolocation correction. By integrating new criteria, including RH profile and terrain matching, and utilizing parallel processing methods, the framework overcomes the limitations of existing methods like the GEDI Simulator. *GEDICorrect* demonstrates superior performance across all tests, positioning it as a more viable and essential tool for accurate vegetation monitoring and ecosystem assessment.

Keywords: Parallel Programming, Geolocation Correction, Remote Sensing, Simulation, Data Processing

Sumário

Melhoria da Precisão do GEDI Combinando Diferentes Critérios de Correção de Geolocalização e Programação Paralela

À medida que os desafios ambientais globais se intensificam, a monitorização dos ecossistemas tornou-se crucial para enfrentar as alterações climáticas. As missões LiDAR espaciais, como o Global Ecosystem Dynamics Investigation (GEDI) da NASA, desempenham um papel fundamental na quantificação e monitorização da estrutura tridimensional da vegetação. O GEDI fornece medições de alta resolução da estrutura e topografia da floresta, mas essas leituras são frequentemente afetadas por erros de geolocalização causados pela instabilidade da plataforma e interferência atmosférica, comprometendo a precisão das estimativas da altura da vegetação e do solo. Os métodos existentes de correção de geolocalização, como o GEDI Simulator, calculam o erro médio (em metros) da geolocalização dos *footprints* por órbita, o que se tem vindo a mostrar inadequado para paisagens heterogéneas.

Esta dissertação apresenta o *GEDICorrect* como uma nova abordagem para a correção de geolocalização à escala do *footprint*. Ao integrar novos critérios, e utilizar métodos de programação paralela, a abordagem proposta supera as limitações dos métodos existentes, como o *GEDI Simulator*. O *GEDICorrect* demonstrou um desempenho superior em todos os testes, tornando-o como uma ferramenta robusta e essencial para a monitorização da vegetação e avaliação dos ecossistemas.

Palavras chave: Programação Paralela, Correção Geolocalização, Deteção Remota, Simulação, Processamento de Dados

1

Introduction

As environmental concerns continue to intensify, the need for effective monitoring of terrestrial ecosystems has become more urgent. Remote sensing technologies, particularly spaceborne LiDAR, have emerged as essential tools for evaluating changes in forest structure and land cover. These technologies provide critical insights that support conservation efforts and contribute to climate change mitigation strategies.

1.1 Context

Terrestrial ecosystems - such as forests, shrublands, grasslands, and wetlands - are essential components of the Earth's biosphere [Kyker-Snowman et al., 2021]. These ecosystems provide critical services, including carbon sequestration, nutrient and water cycling, and biodiversity conservation, all of which are vital to human well-being and the functioning of the planet's natural systems [Chapin et al., 2011]. Despite their importance, terrestrial ecosystems are increasingly threatened by human activities such as deforestation, land-use change, and climate change. Understanding the impacts of rapid changes in the extent and structure of these ecosystems on climate, habitats, and biodiversity is crucial for developing effective conservation and mitigation policies.

One key aspect of monitoring terrestrial ecosystems is quantifying three-dimensional (3D) vertical vegetation structure. Parameters such as vegetation height, canopy cover, density, and heterogeneity are crucial for many ecosystem processes and modeling studies (e.g., [Guo et al., 2021]). For instance, vegetation height is a fundamental variable for: i) estimating aboveground biomass (AGB), which is essential for assessing and modeling global carbon fluxes [Lefsky et al., 2005, Simard et al., 2011]; ii) assessing and characterizing habitat structural heterogeneity, an important factor in explaining biodiversity spatial patterns [Bergen et al., 2009, Carrasco et al., 2019]; iii) improving the accuracy of microclimate condition estimates, such as temperature, humidity, and radiation regimes [Zellweger et al., 2019, De Frenne et al., 2021]; and iv) supporting fire management activities, as vegetation height is a key input for fire spread simulations [Saatchi et al., 2007]. Thus, accurately monitoring and understanding the complexity of terrestrial ecosystem processes, dynamics, and vulnerabilities, as well as developing effective management strategies, largely depends on the availability of timely, high-resolution data on 3D vegetation structure parameters, such as vegetation height (e.g., [Hall et al., 2011]).

While field-based measurements can provide accurate estimates of 3D vegetation metrics, these methods are time-consuming, labor-intensive, and limited in their ability to provide spatially continuous information over large areas. Satellite remote sensing offers advanced technology with the potential to deliver vegetation vertical metrics effectively, systematically, and consistently on a large scale [Szpakowski and Jensen, 2019]. Light Detection and Ranging (LiDAR) and Synthetic Aperture Radar (SAR) have shown strong capabilities for estimating vegetation structure parameters due to their sensitivity to surface structure [Pardini et al., 2019]. LiDAR, in particular, has emerged as a widely used technology for acquiring three-dimensional information, providing high-accuracy data on vegetation structure [Dong and Chen, 2017].

LiDAR systems emit laser pulses (circular pulses, or footprints) that can penetrate vegetation canopies through gaps between leaves and branches, enabling accurate estimation and reconstruction of vertical information and the internal structure of vegetation canopies [Moudrý et al., 2022]. Spaceborne LiDAR sensors, mounted on satellites, extend these capabilities further by providing accurate measurements of the Earth's surface and vegetation structure on a global scale. In 2018, NASA launched two significant spaceborne LiDAR missions: the Ice, Cloud, and Land Elevation Satellite (ICESat-2) [Neumann et al., 2019], and the Global Ecosystem Dynamics Investigation (GEDI) instrument attached to the International Space Station (ISS) [Dubayah et al., 2020]. Both missions have been collecting and delivering extensive LiDAR datasets at a near-global scale, presenting an unprecedented opportunity to assess and estimate key vertical vegetation metrics across large areas, free of cost, and with high temporal frequency [Potapov et al., 2021, Malambo and Popescu, 2024].

However, spaceborne LiDAR data often require correction due to various sources of error, including instrument inaccuracies, atmospheric conditions (e.g., dense cloud cover), and spacecraft platform instability [Xu et al., 2023]. One of the primary challenges in using spaceborne LiDAR data, particularly from GEDI, is the geolocation errors associated with the measurements [Tang et al., 2023, Ruoqi Wang and Li, 2024]. The reported coordinates may not represent the exact location of the measurements but rather a nearby location in the surrounding area (see Sections 2.2.2 and 5.2). Currently, GEDI's horizontal geolocation accuracy is approximately 10 meters for calibrated final products [Beck et al., 2021], which can introduce errors when assessing the accuracy of canopy height and terrain elevation estimates.

Efforts to improve GEDI's geolocation accuracy have been developed and implemented by the scientific community [Hancock et al., 2019, Quirós Rosado et al., 2021, Xu et al., 2023]. Notably, the GEDI Simulator tool, developed by the GEDI Science Team [Hancock et al., 2019], which incorporates the `collocateWaves` program, has been widely used to reduce geolocation errors in GEDI data. This approach assumes a systematic error across the orbit [Tang et al., 2023] and aims to find a coordinate offset to apply to the entire orbit to correct horizontal deviations (see Section 3.1). However, the assumption of a uniform system-

atic error across the orbit is likely too optimistic and may not hold true [Tang et al., 2023]. As a result, footprint-level correction methods have been implemented, where the offset is calculated for each individual footprint rather than for the entire orbit (e.g. [Quirós Rosado et al., 2021]).

1.2 Motivation

Despite being NASA's official tool for addressing geolocation errors in GEDI data, the GEDI Simulator presents several limitations that hinder its practicality and reliability. One significant barrier is the complexity of running the program, which may be inaccessible to many remote sensing scientists due to a lack of user-friendly documentation and interfaces. Additionally, the GEDI Simulator is inefficient, consuming excessive memory and operating slowly, making it unsuitable for large-scale applications. Moreover, using an orbit-level correction method, it lacks the precision needed for individual footprints, reducing its effectiveness in areas with high land cover heterogeneity. These limitations have motivated the creation of ***GEDICorrect***, a framework designed to enhance geolocation accuracy at the footprint-level, which represents the main contribution of this study. The development of this framework was driven by the need for a more efficient, accurate, and scalable geolocation correction method that can handle large datasets while leveraging parallel processing techniques, and introduces new methods, criteria, and metrics for improving footprint geolocation accuracy. By enhancing geolocation precision, this framework enables a better assessment of canopy structure, that can be applied to a wide range of fields, from advancing our understanding of carbon sequestration to supporting more informed planning and conservation efforts.

1.3 Objectives

The main goal of this study is to develop and test a new footprint-level correction approach, to improve GEDI geolocation accuracy. To achieve such goal, the following specific objectives were defined:

1. Assess the effectiveness of the standard geolocation correction process (`collocateWaves`) in improving the accuracy of the GEDI footprints (Baseline Assessment);
2. Introduce new criteria and assess how different combinations improve the accuracy of footprint geolocation (Criteria Assessment);
3. Leverage parallel processing to optimize the GEDICorrect framework and evaluate its efficiency (Efficiency Assessment);
4. Evaluate the trade-offs between the number of simulated points and computational cost, and assess the impact of randomness in point generation on geolocation accuracy (Points Distribution Assessment).

Ultimately, the proposed framework will serve as a valuable tool for ecosystem monitoring, providing insights into vegetation structure and land surface elevation, both of which are crucial for global environmental assessments.

1.4 Contributions

This work makes several key contributions to the field of GEDI data processing and geolocation correction. The main contribution is the design and implementation of this dissertation's proposed framework,

GEDICorrect, which enhances the geolocation accuracy of GEDI data. The second main contribution is the introduction of new methods, criteria, and metrics specifically designed to address limitations of existing geolocation correction techniques, such as GEDI Simulator, by incorporating them in the newly implemented solution. By focusing on correction at the footprint-level, rather than orbit-level, this study offers a fine-grained alignment of GEDI data with actual ground readings. The flexibility of *GEDICorrect* allows for multiple correction strategies, making it adaptable to different landscape types and measurement requirements, supporting applications in vegetation structure analysis, biomass estimation, and canopy height modeling.

Before developing *GEDICorrect*, a review of current GEDI geolocation correction methods and tools was made. This review synthesizes the state-of-the-art approaches and provides the foundation for this study. The analysis of existing methods reveals gaps in precision and adaptability, which *GEDICorrect* addresses.

Furthermore, this work contributes a merged dataset combining GEDI L1B and L2A data products, referred to as *GEDI_CorrectTest*. This dataset was created for the experiments in this research, providing a consistent testing environment for the proposed metrics, such as vegetation profile characteristics, terrain elevation, and GEDI waveform properties. This dataset lays the groundwork for a new approach to GEDI data analysis by integrating different data levels to improve geolocation correction.

Finally, a comparative analysis was conducted to assess the performance of existing correction methods (e.g., GEDI Simulator) against *GEDICorrect*. This analysis proves the superiority of the new framework with its improved accuracy and adaptability, highlighting its potential for broad applications in Earth observation and remote sensing research.

1.5 Dissertation Structure

This dissertation consists of six main chapters, with their organization designed to highlight the value of the newly proposed approach in improving GEDI geolocation accuracy. The chapters are organized as follows:

- Chapter 1 (this chapter) provides a general introduction, including the context for the research, motivation, main objectives, and the overall structure of the document;
- Chapter 2 presents the background of the key topics relevant to this work. It includes a brief overview of remote sensing technology, a description of the GEDI spaceborne LiDAR sensor, and an introduction to the concept of parallelization;
- Chapter 3 delves into the design and functionality of the primary frameworks for GEDI geolocation correction. First, the existing GEDI Simulator's geolocation correction methods are analyzed to establish a performance benchmark. Then, the newly proposed GEDICorrect framework is introduced, offering footprint-level correction methods and leveraging parallel processing to enhance performance and scalability;
- Chapter 4 outlines the methodology, detailing the integration of the frameworks with the dataset and the experimental setup. It begins by presenting the study area and data collection methods, followed by a description of the use of both frameworks. Finally, a series of experiments are defined to evaluate the performance of GEDICorrect's geolocation correction methods;
- Chapter 5 presents the main results from each experiment and discusses them in comparison with existing literature on the subject;

- Chapter 6 summarizes the main conclusions of the comparative analysis between GEDI Simulator and the proposed GEDICorrect framework in terms of accuracy and efficiency. The chapter concludes with suggestions for future research.

2

Background

This chapter presents the theoretical foundations and definitions used throughout the dissertation. The content is structured to provide an overview to remote sensing technologies such as Airborne Laser Scanning (ALS) and the GEDI spaceborne LiDAR mission. Additionally, it introduces the concepts of parallel processing that were fundamental to the development and optimization of the proposed framework.

2.1 Remote Sensing

Satellite-based Earth observation involves studying the Earth and its environment through remote sensing techniques. Remote Sensing is a technique that allows identifying, measuring, and observing objects, areas, or phenomena, without direct contact, typically through the analysis of data acquired by sensors located remotely [Schott, 2007, Lillesand et al., 2015]. Earth-orbiting satellites equipped with electromagnetic sensors collect data by detecting the electromagnetic radiation emitted or reflected by surface features. This data is then analyzed to derive valuable information about the areas or phenomena being studied. While field-based data collection and near-surface instrumentation can provide valuable insights into natural processes, such as ecosystem structure, functions and dynamics, these methods are often time-consuming,

labor-intensive, and limited in their ability to deliver spatially continuous information over large areas. Remote sensing enables us to obtain a broad view of the Earth at varying spatial and temporal scales for monitoring environmental changes, studying vegetation dynamics, and addressing challenges related to climate change, land cover changes, and biodiversity conservation.

Sensors in remote sensing are capable of recording objects through Electromagnetic Radiation (ER) across a wide range of wavelengths far beyond the visible spectrum (Figure 2.1). This capability is achieved by measuring the ER reflected or emitted from an object. The interaction between this radiation and the object - whether it is absorbed, transmitted, or reflected - provides crucial information about the object's properties [Lillesand et al., 2015]. The collection of these electromagnetic readings depends on the type of sensor used, each with its unique method of acquiring information.

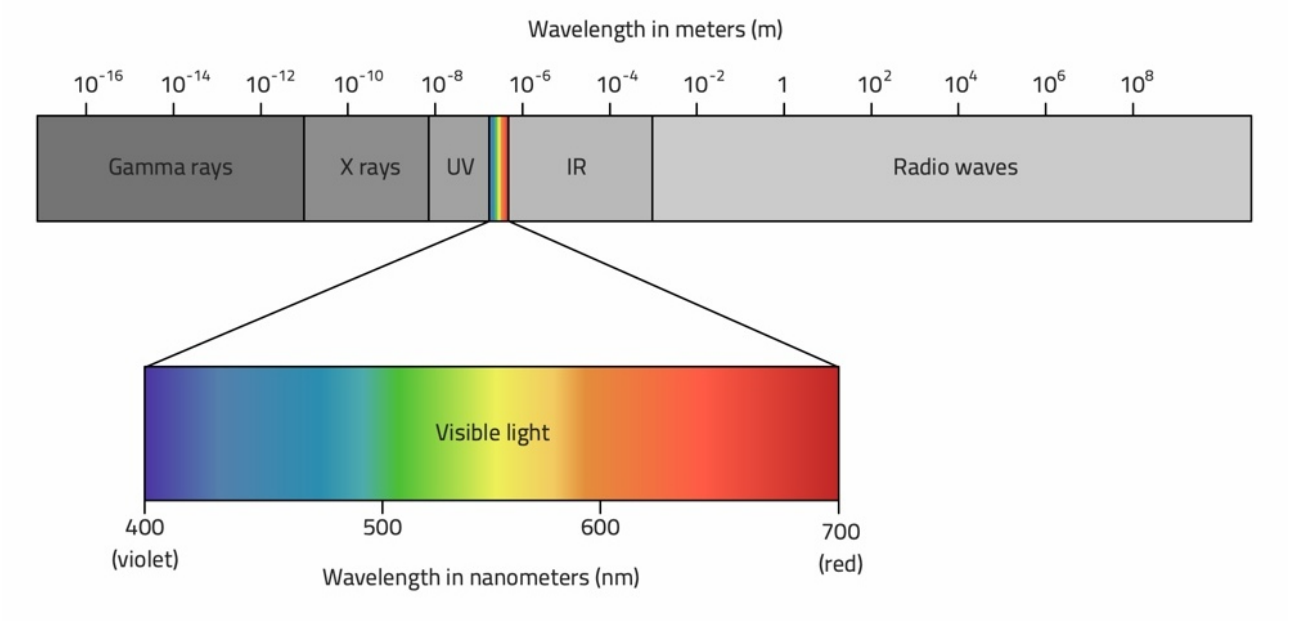


Figure 2.1: The Electromagnetic Spectrum [Samvedan, 2020]

There are two main types of sensors in remote sensing:

- **Passive Sensors** - which detect natural radiation emitted or reflected by objects, typically using sunlight as the source of illumination.
- **Active Sensors** - which emit their own signals (such as laser pulses or radar waves) and measure the reflection from the target, allowing precise measurements even in the absence of external light sources.

Remote sensing sensors, whether passive or active, operate across different regions of the electromagnetic spectrum. Passive sensors primarily capture energy within the visible, infrared, and thermal regions of the spectrum, while active sensors often operate in the microwave and infrared regions. LiDAR systems, for instance, normally uses laser pulses in the near-infrared range (typically around 1064 nm) and/or green ($\simeq$532 nm) to generate highly accurate 3D models of terrain and vegetation [Cracknell, 2007]. Active sensors such as airborne and spaceborne LiDAR systems, which excel in generating detailed 3D representations of the Earth's surface, will be the focus of this dissertation.

Remote sensing instruments can be mounted on a variety of platforms, from ground-based stations to aircraft and satellites (Figure 2.2). Airborne Laser Scanning (ALS) systems, for example, are LiDAR

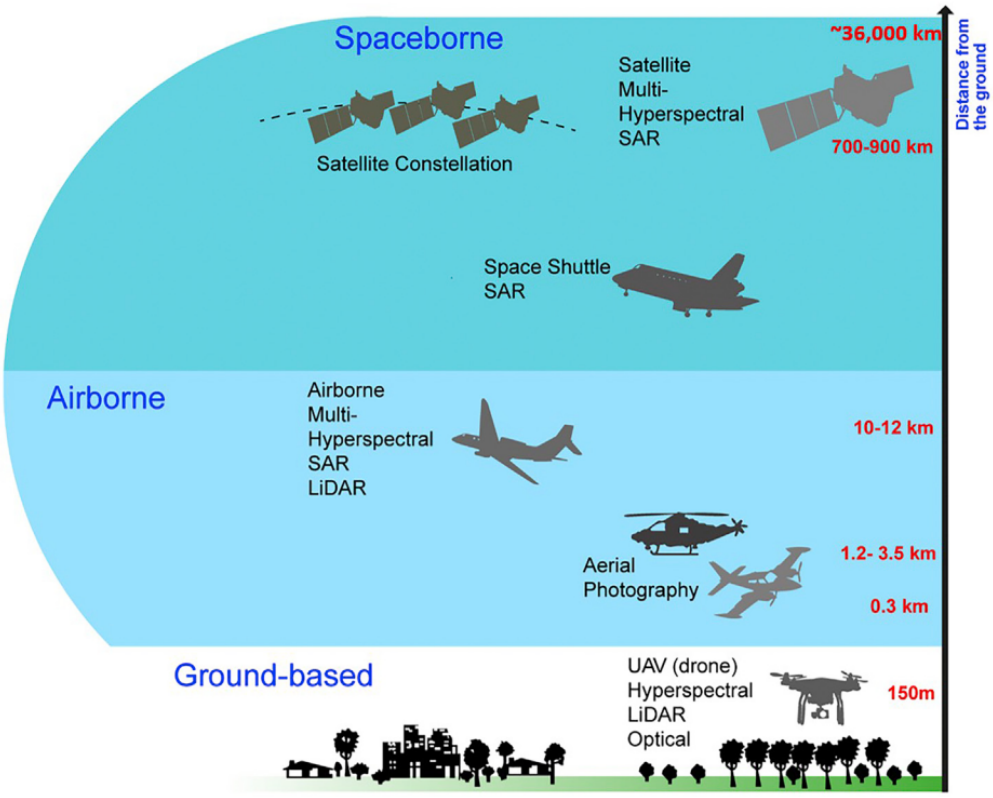


Figure 2.2: Main remote sensing instrument platforms [Lechner et al., 2020]

sensors installed on aircraft that fly over specific terrestrial ecosystems to capture high-resolution 3D data. On the other hand, spaceborne LiDAR technologies, such as GEDI [Dubayah et al., 2020] and ICESat-2 [Abdalati et al., 2010], provide global-scale LiDAR data, albeit with lower resolution compared to airborne systems. The choice of the platform involves a trade-off between coverage, resolution and cost. Airborne LiDAR systems deliver highly detailed and accurate data but are limited to smaller geographic areas due to high acquisition costs [Gwenzi et al., 2016], while spaceborne LiDAR offers near-global coverage in high spatial resolutions.

Regardless of the platform, remote sensing data must be accurately georeferenced for analysis. This is done using a Coordinate Reference System (CRS), which defines how spatial data is projected onto a map [Chang, 2018]. The most widely used CRS is the World Geodetic System 1984 (WGS84)[1], which expresses its coordinates in latitude, longitude, and ellipsoidal height. Additionally, specific studies may employ different systems for improved precision. For instance, certain geodetic applications rely on EPSG[2] Geodetic Parameter Datasets, which is a public registry of spatial reference systems and related units of measurement. Each entity is assigned an EPSG code between 1024 and 32767[3]. For example, the CRS with code EPSG:3041 is a map projection used for datasets covering regions like Portugal. Once georeferenced, this data can be explored and analyzed using Geographic Information Systems (GIS) software, such as QGIS[4], which was used throughout this work for data visualization and processing during the geolocation correction process.

[1] https://svenruppert.com/2023/12/18/what-is-wgs84-an-overview/
[2] European Petroleum Survey Group
[3] https://proceedings.esri.com/library/userconf/petrol13/papers/petrol_10.pdf
[4] https://www.qgis.org/

2.1.1 Lidar Principles and Applications

Given the strong focus of this dissertation on remote sensing LiDAR technology, this section provides a brief introduction to the topic. However, it does not aim to offer the comprehensive and exhaustive characterization that such technology warrants. For a more in-depth analysis, key publications in the field (e.g., [Dong and Chen, 2017]) should be consulted. In general, LiDAR instruments measure the distance between the sensor and a target surface by calculating the time interval between the emission of a laser pulse and the detection of its reflection, known as the return signal, at the sensor's receiver [Bachman, 1979]. This time interval is used to compute the round-trip distance, which, when halved, gives the actual distance to the target. The distance is derived using the speed of light, as shown in Equation 2.1:

$$d = \frac{(c \times t_{elapsed})}{2} \tag{2.1}$$

where c is the speed of light ($c = 299\,792\,458\ m/s$) and $t_{elapsed}$ is the time interval between emission of laser pulse and return signal. The laser pulse is assumed to be circular, which is also called the *laser footprint* [Wehr and Lohr, 1999].

The key characteristics among LiDAR instruments [Lefsky et al., 2002] are related to:

- **Laser's Wavelength** - measured in nanometers, usually emitted between 900–1064 nanometers for terrestrial applications, which is part of the near-infrared spectrum. One of the major drawbacks of using these wavelengths is that clouds absorb the signals, making it difficult to use the devices in cloudy conditions [Lefsky et al., 2002]. For bathymetric sensors, the emitted wavelength ranges near 532 nm for better penetration of water [Irish and White, 1998];
- **Power** - determines how far the pulse can travel and still return a detectable signal, affecting the range and ability to penetrate dense vegetation [Wehr and Lohr, 1999];
- **Pulse Duration** - refers to the length of time the laser emits a single pulse (measured in ns);
- **Repetition Rate** - explains how frequently the laser emits pulses, with higher repetition rates resulting in denser point clouds (measured in Hz);
- **Beam Size** - measured in meters, defines the area covered by each laser pulse on the ground (footprint size). The higher in altitude the instrument is, the larger the footprint [Lim et al., 2003];
- **Divergence Angle** - is the spread of the laser beam as it travels on the instrument's platform.

Among these, the **beam size** is especially significant when differentiating between small-footprint and large-footprint LiDAR systems. Small-footprint LiDAR measurements typically cover an area of less than 1 meter in diameter and is commonly used in Drone-mounted and ALS systems, resulting in highly detailed point clouds. In contrast, large-footprint LiDAR, which can be mounted on aircraft at higher altitudes or used in spaceborne missions, cover footprint areas ranging from 10 to 100 meters in diameter. This wider beam size allows large-footprint systems to capture broader patterns over extensive areas, making them suitable for vegetation structure analysis at a large scale.

When mounted on an aircraft, ALS systems rely on precise navigation and positioning technology to ensure that each laser pulse is accurately geolocated, which is achieved through a combination of GPS (Global Positioning System) and an IMU (Inertial Measurement Unit). The GPS provides spatial coordinates, while the IMU accounts for the aircraft's roll, pitch, and yaw during flight. Together, these systems allow for the precise geolocation of each LiDAR point [Lefsky et al., 2002], as demonstrated in Figure 2.3.

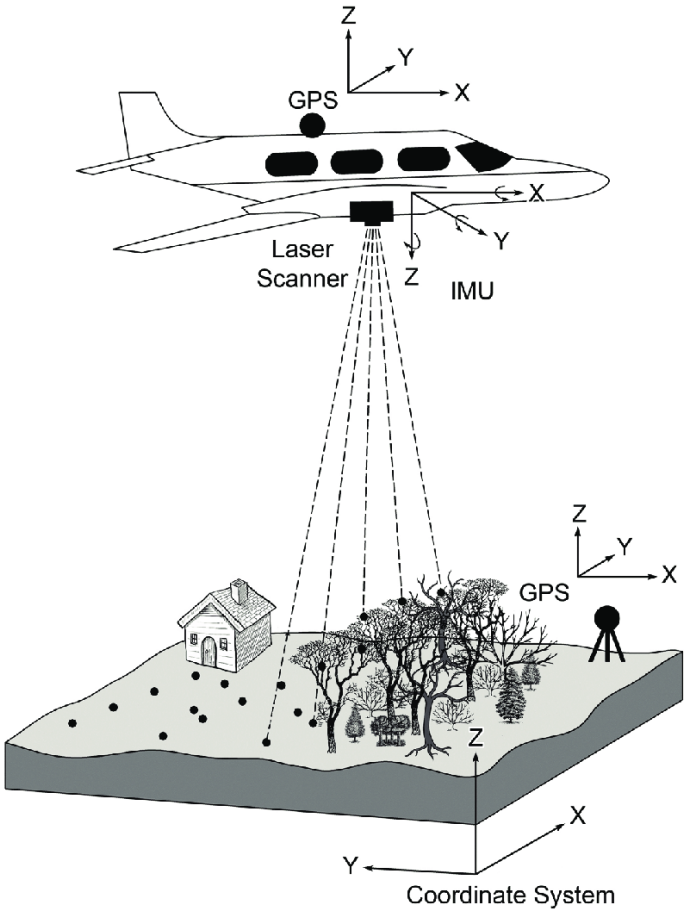


Figure 2.3: Typical Airborne Laser Scanning system [Roman and Ursu, 2016]

In addition to the differences in footprint size, LiDAR systems can also be categorized based on how they capture and process the reflected laser energy: i) Discrete-return; ii) Full Waveform; and iii) Photon Counting [Sumnall et al., 2016, Mandlburger et al., 2019] (Figure 2.4).

In discrete-return LiDAR, the instrument records a limited number of reflection points per laser pulse (typically 1 to 5), corresponding to high-intensity reflections from different surfaces across the vertical vegetation axis. Each return is typically categorized into classes such as canopy top, intermediate vegetation layers, or the ground. This method is the most widely used in ALS systems, which commonly produce small-footprint data [Sumnall et al., 2016]. In contrast, full waveform-return LiDAR captures the full energy profile of the laser pulse as it reflects back to the sensor. Instead of recording discrete points, it digitizes the entire returning waveform at regular intervals, providing continuous vertical information across the footprint. This technique allows for a more detailed characterization of the vertical structures [W. Wagner and Ducic, 2008]. Finally, photon-counting LiDAR systems detects individual photons within each laser pulse, often emitted in a grid pattern across the surveyed area. Due to its high sensitivity and ability to operate with low-energy pulses, photon-counting LiDAR can achieve greater range and broader coverage [Mandlburger et al., 2019]. However, this method is also highly susceptible to noise from ambient light or atmospheric conditions [Jiang et al., 2023]. An example of this technology is the ATLAS sensor from the ICESat-2 mission, which employs photon-counting LiDAR [Smith et al., 2019].

Typically, the processing team of a company conducting an ALS scan over an area provides the point-cloud data in `.las` files. A `.las` file is a standardized binary file format used to store LiDAR data, containing

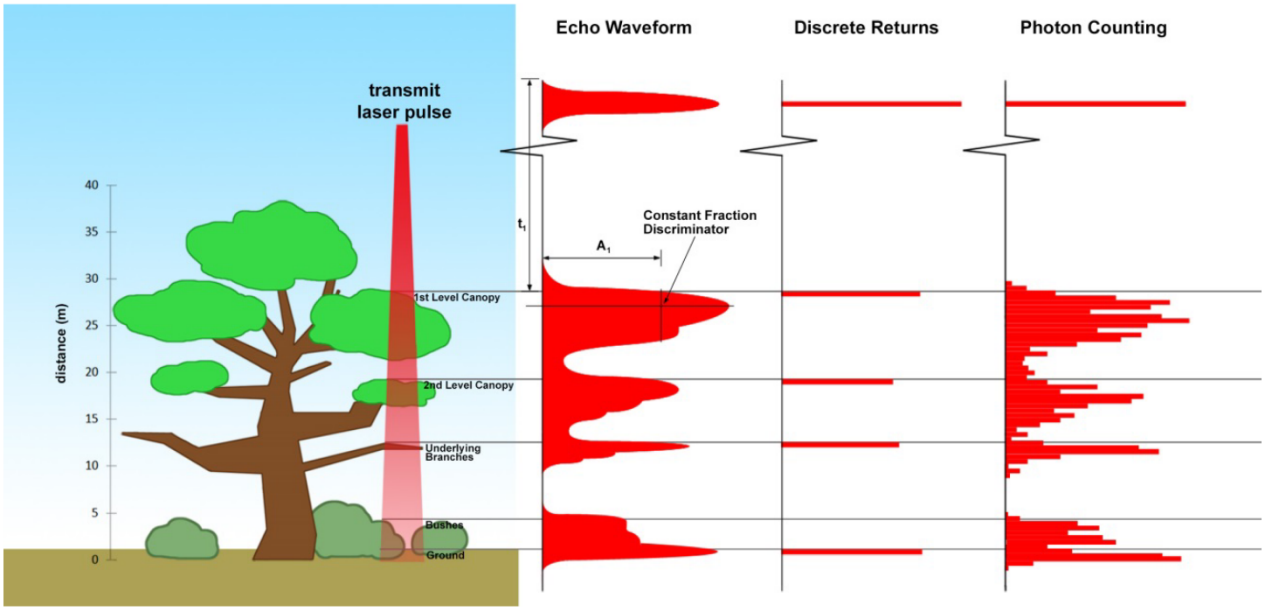


Figure 2.4: Various modalities of LiDAR detection [Neuenschwander et al., 2023]

detailed 3D coordinates (X, Y, Z) of laser return points, along with additional attributes like intensity, return number, classification, GPS time, and color information[5].

2.2 GEDI

The Global Ecosystem Dynamics Investigation (GEDI) is the first spaceborne LiDAR technology specifically designed to globally measure and monitor the three-dimensional structure of the vegetation and topography, providing crucial insights into Earth's carbon storage, ecosystem structure, and biodiversity [Dubayah et al., 2020]. Successfully launched from Cape Canaveral, GEDI was carried in the Dragon capsule of SpaceX CRS-16 on a Falcon 9 rocket and subsequently installed in the Japanese Experiment Module-Exposed Facility (JEM-EF) on the ISS in December 2018. GEDI measurements are conducted day and night, continuously covering the Earth's land surfaces between 51.6° N and 51.6° S latitudes, encompassing the tropical and temperate forests of the Earth. As with all optical remote sensing, GEDI observations cannot be made through dense cloud cover [Lefsky et al., 2002, Dubayah et al., 2020]. The sensor employs a system with three main lasers, producing eight parallel beams (four "coverage" beams and four "full power" beams) for surface readings. These beams illuminate an area on the Earth's surface equivalent to a circle of approximately 25 meters in diameter, known as the **footprint**. The laser sensors used by GEDI are the High Output Maximum Efficiency Resonator (HOMER) with pulse length of 15.6 ns and pulse repetition rate of 242 Hz, emitting a laser beam at wavelength of 1064 nm (near-infrared) [Stysley et al., 2015, Duncanson et al., 2020], allowing for three-dimensional measurements of the surface. The distance between each footprint center is about 60 meters along the flight direction, and they are spaced approximately 600 meters across the track direction from each other (see Figure 2.5).

Due to the ISS not having a regular orbit, GEDI does not guarantee a revisit cycle for new acquisitions in the same location [Dubayah et al., 2020]. Figure 2.6 demonstrates an example of GEDI track coverage.

In GEDI, an onboard telescope collects and records light reflected from the ground, vegetation, and, in

[5] https://www.loc.gov/preservation/digital/formats/fdd/fdd000418.shtml

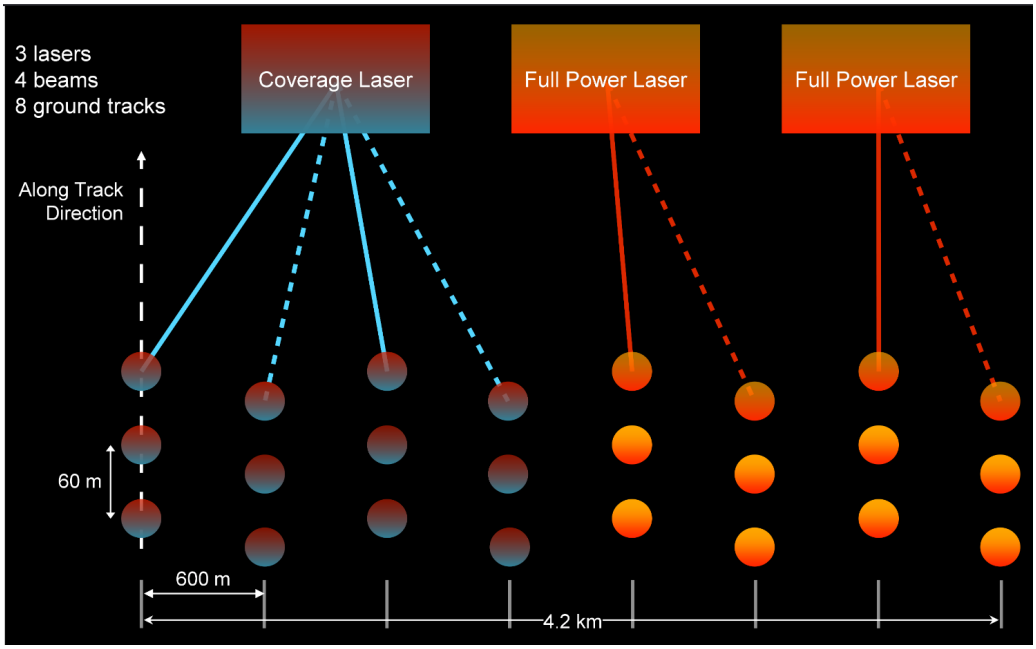


Figure 2.5: GEDI ground sampling pattern. The circles represent the GEDI footprints (with a diameter of 25 meters) https://gedi.umd.edu/instrument/specifications/.

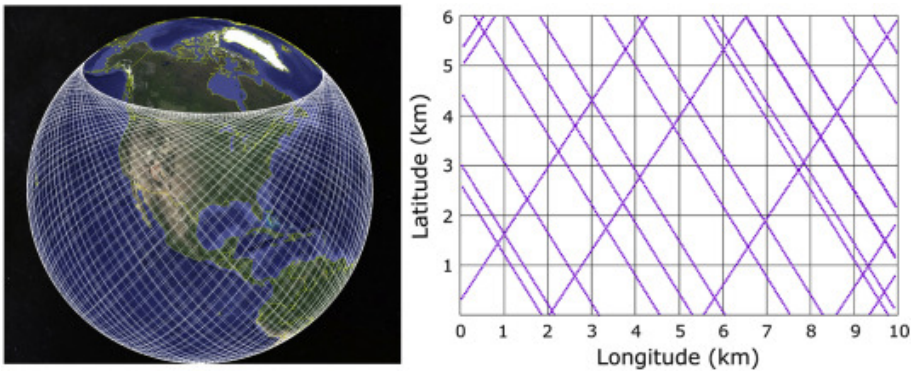


Figure 2.6: Illustration of the GEDI coverage (left) and an example of a GEDI track on the Equator (right). Adapted from [Dubayah et al., 2020].

many cases, clouds. The recorded light, representing the laser energy reflected by the surface of objects within the footprint at different heights, is converted into voltage and recorded over time in 1 nanoseconds (ns) intervals. Following this conversion, the height of objects is calculated by multiplying the recorded time and the speed of light, producing the full waveform [Fayad et al., 2020]. Figure 2.7 demonstrates this, the left plot of the figure represents a GEDI LiDAR full-waveform example. The orange area beneath the curve represents the energy reflected back from the canopy, while the darker brown portion indicates the return signal from the underlying terrain. The black line records the cumulative return energy, spanning from the base of the ground return (normalized to 0) to the top of the canopy (normalized to 1). Relative Height (RH) metrics provide insights into the height at which a specific quantile of the returned energy is reached concerning the ground (center of the ground return). The accompanying diagram on the right of Figure 2.7 visually represents the distribution of trees responsible for generating the waveform on the left.

From the waveform recorded in each footprint, a set of vertical structure metrics can be derived [Drake et al., 2002, Tang et al., 2012]. These include vegetation canopy height, canopy cover, plant

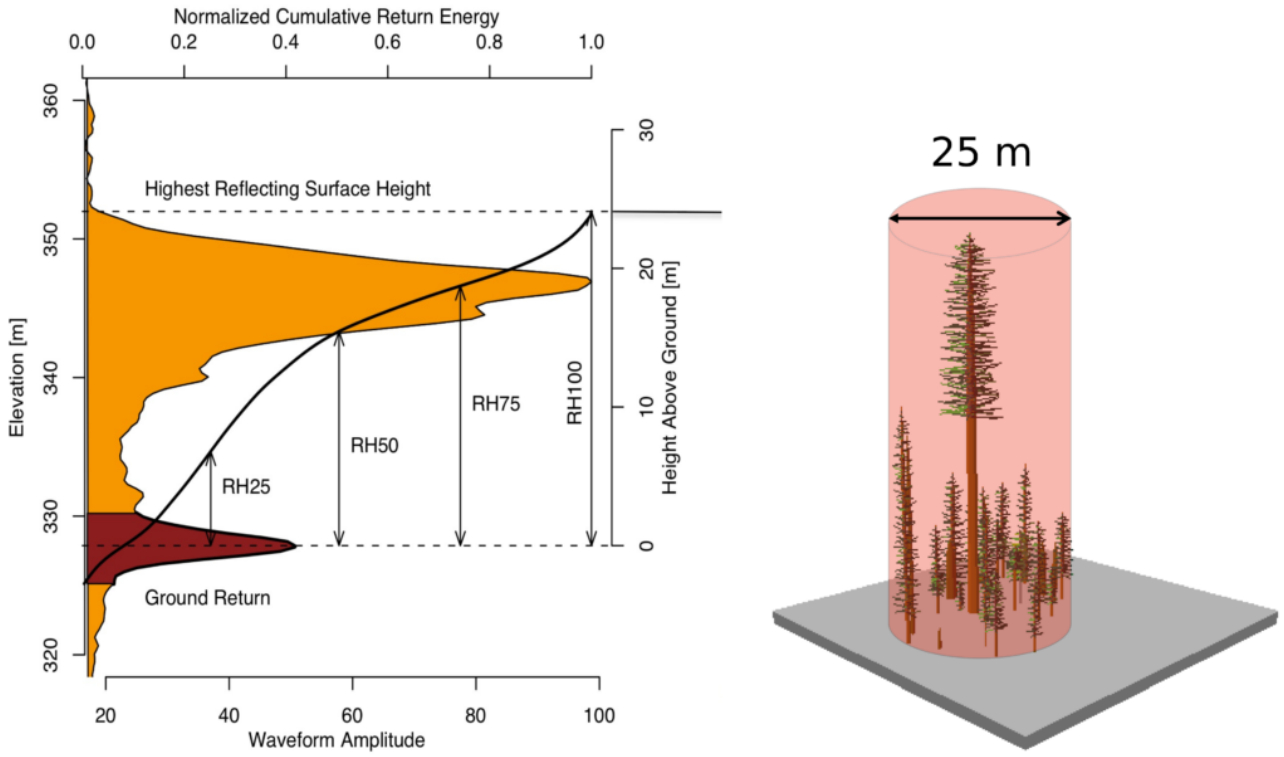


Figure 2.7: Representation of a GEDI LiDAR full-waveform and its respective distribution of trees responsible for generating such waveform. Adapted from https://gedi.umd.edu/data/products/.

area index and vertical foliage profiles, topography, as well as footprint-level and gridded Aboveground Biomass Density (AGBD) besides others [Dubayah et al., 2020].

Specifically, GEDI offers seven main scientific data products, which encompass:

- **L1A Raw Waveforms** - Level 1A product contains fundamental instrument engineering and housekeeping data as well as the raw waveforms of each footprint of 25m diameter and geolocation information, used to compute higher level data products;
- **L1B Geolocated Waveforms** - Level 1B product provides geolocated corrected and smoothed waveforms from L1A data. The L1B product contains 85 layers for each of the eight beams, including the geolocated and smoothed waveform datasets and parameters;
- **L2A Geolocated Elevation and Metrics** - contains waveform interpretation metrics from each L1B received waveform, including ground elevation, canopy top height and relative height (RH) metrics. The L2A product contains 156 layers for each of the eight beams, including the metrics described previously;
- **L2B Canopy Cover and Vertical Profile Metrics** - this product includes extracted biophysical metrics from each GEDI waveform. These metrics are based on the directional gap probability profile derived from the L1B waveform. Metrics provided include canopy cover, Plant Area Index (PAI), Plant Area Volume Density (PAVD), and Foliage Height Diversity (FHD);
- **L3 Gridded Level 2 Metrics** - This dataset provides GEDI Level 3 (L3) gridded mean canopy height, standard deviation of canopy height, mean ground elevation, standard deviation of ground elevation, and counts of laser footprints per 1km x 1km grid cells globally within -52 and 52 degrees latitude;

- **L4A Footprint Level AGBD** - this product contains GEDI Level 4A (L4A) Version 2 predictions of the aboveground biomass density (AGBD; in Mg/ha) and estimates of the prediction standard error within each sampled geolocated laser footprint;
- **L4B Gridded AGB Density** - provides 1 km x 1 km estimates of mean AGBD based on observations from 2019-04-18 to 2023-03-16. The GEDI L4A Footprint Biomass product converts each high-quality waveform to an AGBD estimate, while the L4B product uses the sample present within the borders of each 1 km cell to statistically infer mean AGBD.

One of the most crucial metrics included in GEDI Level 2 data products is the RH profile. The RH profile represents the height distribution at which a specific percentage of the returned laser energy is reflected by the vegetation and/or ground surfaces, ranging from RH0 (the ground level) to RH100 (the highest detected return). For example, an RH50 of 7.5 meters indicates that 50% of the total energy from the laser pulse was reflected by the surfaces below 7.5 meters. This metric provides insight into the vertical structure of the vegetation canopy, helping in estimating canopy height and biomass.

The GEDI instrument was specifically engineered to measure vertical canopy profiles in conditions with canopy cover reaching up to 95% and 98% for the coverage and power beams, respectively. Its design was fine-tuned for optimal performance in measuring dense forests, drawing on over two decades of research utilising airborne large footprint waveform LiDAR [Drake et al., 2002, Dubayah et al., 2020]. With a short pulse length (Full Width Half Maximum, FWHM, of 15.6 ns), GEDI can effectively discriminate between canopy and ground returns in forested ecosystems. However, it's important to note that the instrument's design did not prioritise the characterization of short stature and discontinuous vegetation, although it could offer valuable and accurate insights into the characterization of lower stature and discontinuous vegetation [Li et al., 2023, Zhu et al., 2023].

2.2.1 Factors affecting GEDI canopy height estimates

Since GEDI is a spaceborne lidar system, its measurements are subject to several factors that can affect the accuracy of canopy height estimates. Being mounted on the ISS, GEDI collects data from a higher altitude compared to ALS systems, leading to greater susceptibility to geolocation errors and signal distortions. Overall, there are seven main factors that can affect GEDI signal and, consequently, the accuracy of key derived products, such as canopy height and terrain elevation (in the L2A product). These factors include:

- **GEDI pre-processing algorithms** - GEDI employs 6 different pre-processing algorithms to derive L2A and L2B ground and vegetation metrics from the L1B received waveforms [Hofton et al., 2019]. These algorithms, collectively known as algorithm setting groups (SG), represent specific parameter values for both smoothing and threshold settings used to interpret the received waveform under various conditions (Figure 2.8). For instance, the results of the linear models comparing the on-orbit GEDI canopy height measurements and ALS-derived canopy height were found to range varyingly, depending on the algorithm employed [Lahssini et al., 2022];
- **Slope** - Terrain slope has a strong impact on large footprint LiDAR systems, where steep terrains can broaden the LiDAR waveform, thereby affecting the accurate extraction of canopy heights [Adam et al., 2020, Dhargay et al., 2022, Fayad et al., 2021, Wang et al., 2022]. It has been reported that in complex terrain slopes (e.g. with slope >20%) the bias in canopy height estimations significantly increases [Fayad et al., 2021];
- **Canopy cover and height** - GEDI exhibits its highest canopy height accuracy when both canopy cover and height are within moderate levels; in low-canopy conditions the waveform energy is more

likely to be reflected from the terrain surface rather than the canopy and vice-versa for high-canopy conditions [Dhargay et al., 2022, Dorado-Roda et al., 2021, Zhu et al., 2022];

- **Acquisition time (day or night time)** - it is anticipated that measurements taken during the day are less accurate than those conducted during the night due to additional radiance signal from sunlight that could scatter into the GEDI telescope from the atmosphere and surface [Fayad et al., 2022];
- **Beam Type** - In dense forests, coverage beams, having less energy to penetrate towards the ground when compared to power beams, are expected to yield inferior performance. Hence, the use of power beams is generally recommended, even in night time conditions [Beck et al., 2021, Zhu et al., 2022];
- **Sensitivity** - Beam sensitivity, which can be interpreted as the GEDI's penetrating capability for ground detection, is affected by the strength of GEDI return signals and canopy cover, so an effect is expected on heights measured especially over areas of dense forests, commonly seen in the tropics [Hofton et al., 2019, V.C. Oliveira et al., 2023, Hancock et al., 2019].
- **Geolocation Errors** - Geolocation errors in spaceborne LiDAR systems such as observed in GEDI, can compromise the linkage between GEDI-measured vegetation height and/or terrain elevation with ALS or field-based measurements (e.g. [Shannon et al., 2024]). In fact, geolocation errors is considered one of biggest challenges in using GEDI data [Ruoqi Wang and Li, 2024].

Algorithm Setting Group (SG)	Smoothing Width (Noise)	Smoothing Width (Signal)	Waveform Signal Start Threshold	Waveform Signal End Threshold
1	6.5 σ	6.5 σ	3 σ	6 σ
2	6.5 σ	3.5 σ	3 σ	3 σ
3	6.5 σ	3.5 σ	3 σ	6 σ
4	6.5 σ	6.5 σ	6 σ	6 σ
5	6.5 σ	3.5 σ	3 σ	2 σ
6	6.5 σ	3.5 σ	3 σ	4 σ

Figure 2.8: List of algorithm setting groups with their corresponding threshold and smoothing values used for interpreting the received GEDI waveform. σ represents the standard deviation of the background noise in the received waveform.

2.2.2 GEDI Geolocation Correction Methods

Numerous studies evaluating the accuracy of GEDI data report a root mean square error (RMSE) ranging from 2.03 m to 10.97 m, with R^2 values between 0.52 to 0.93 when compared with ALS-derived canopy heights [Dorado-Roda et al., 2021, Potapov et al., 2021, Dhargay et al., 2022]. The RMSE quantifies the average magnitude of the errors between predicted and observed values (for specific RH returns) [Hyndman and Koehler, 2006], while R^2 (coefficient of determination) indicates the proportion of variance in the observed data explained by the model, with higher values representing better predictive accuracy [Glantz and Slinker, 1991][6]. These variations depend on the specific forest ecosystems being

[6]The R^2 and RMSE equations are described with detail in Section 4.7

analyzed. However, studies that incorporate geolocation correction consistently show lower RMSE and higher R^2 values, indicating that correcting footprint locations significantly improves the assessment of GEDI accuracy [Wang et al., 2022, Zhu et al., 2022].

Roy et al. [Roy et al., 2021], demonstrated that GEDI's performance is highly sensitive to horizontal geolocation accuracy, with geolocation errors in canopy height estimates being particularly pronounced in areas with high vegetation heterogeneity within GEDI footprints. As of this study, GEDI Version 2 data exhibits a horizontal geolocation accuracy of approximately 10.3 meters (compared to $\simeq$20 meters in Version 1) for calibrated final products, and a vertical accuracy of around 50 cm. Version 3, expected to be released soon, aims to further improve horizontal geolocation accuracy, reducing the error to approximately 8 meters [Dubayah et al., 2020, Beck et al., 2021].

Despite the anticipated improvements in Version 3 data, its utility for many applications remains limited, particularly in arid and semi-arid ecosystems characterized by low-stature, sparse vegetation and high horizontal heterogeneity. In these environments, an 8-meter deviation from the true GEDI measurement locations may hinder the ability to link field-based or ALS-derived AGB data with GEDI AGB estimates (L4 product) for validation tasks. Consequently, there remains a need for not only the continued application of existing geolocation correction methods but also the development and testing of new approaches to further enhance GEDI product accuracy, such as the case of this dissertation.

A substantial body of pioneering and valuable studies has investigated various approaches for validating and correcting geolocation errors in spaceborne LiDAR systems [Luthcke et al., 2001, Sirota et al., 2005, Harding and Carabajal, 2005, Filin, 2006, Magruder et al., 2007, Chunyu et al., 2017, Wang et al., 2020, Zhao et al., 2022, Xu et al., 2023]. For example, Zhao et al. [Zhao et al., 2022] and Xu et al. [Xu et al., 2023] provide clear overviews of this topic, highlighting two primary methods for spaceborne geolocation correction: terrain matching and waveform matching. The terrain matching method corrects footprint location by minimizing the difference between ground elevation derived from a high-resolution Digital Elevation Model (DEM) and the ground elevation reported by the spaceborne LiDAR sensor for that footprint. Specifically, after testing different footprint locations by shifting the footprint along various x and y horizontal coordinates, the position that yields the lowest RMSE between the DEM's true ground elevation and the sensor's reported terrain elevation is selected. The waveform matching method, on the other hand, uses the shape of the waveform to correct geolocation errors by comparing the sensor's reported waveforms with reference waveforms simulated by systems such as ALS. Pearson's correlation between the reported and simulated waveforms has been employed as a criterion [Hofton et al., 2019] to determine the true footprint location. Similar to the terrain matching approach, the objective is to test multiple potential footprint locations and select the one with the highest correlation. This method, the waveform matching using Pearson's correlation, is the one implemented within the *GEDI Simulator* tool.

2.3 Parallel Programming

Parallel programming is a technique that allows multiple calculations or processes to be carried out simultaneously, significantly reducing the runtime of large-scale tasks. In modern multicore systems, this approach is crucial for handling computationally intensive workloads in fields like scientific simulations. As software and hardware evolve, and the number of cores and processors increase, parallelization has become increasingly critical [Grama et al., 2003, Herlihy and Shavit, 2008, Rauber and Runger, 2010]. The evolution of the concept has been driven by the stagnation of CPU clock speeds, leading to the widespread adoption of multicore processor architectures. Consequently, parallel programming is now a necessity for fully utilizing the capabilities of modern hardware across a wide range of applications, from scientific research to everyday business tasks.

However, despite its numerous advantages, parallel programming poses several challenges. It requires a deep understanding of concurrency, memory management, and synchronization mechanisms when common issues arise like race conditions, deadlocks, and data consistency across multiple processes and/or threads. These challenges make parallel programming a complex, but essential, discipline for efficient computation in modern systems.

2.3.1 General Overview

The first step in parallel programming is designing a parallel algorithm or program for a given application. This design begins with the decomposition of a large-scale problem into smaller, independent parts called **tasks**, which can be executed in parallel across the cores or processors of the parallel hardware [Rauber and Runger, 2010].

These tasks are then implemented in a parallel programming language or environment and assigned to processes or threads. This assignment, known as **scheduling**, determines the order in which tasks are executed. Additionally, parallel programs require **synchronization** and coordination of these threads and processes to ensure correct execution. The method by which information is exchanged between processes depends on the task's organization and the memory architecture used in the system.

2.3.2 Serial and Parallel Processes

A serial process refers to a process that is executed by a single core of a single processor, where tasks are carried out one after the other as they appear in the code. In contrast, a parallel process divides a task across multiple cores in a processor or across multiple processors. Each subprocess may operate on its own set of memory while sharing data with other processes as needed. To fully utilize the capabilities of modern multicore systems and supercomputers, parallelization strategies must be employed to distribute workloads efficiently across these resources [Barney and Frederick, 2024].

There is also a distinction between threads and processes. A key distinction in parallel programming lies between multithreading and multiprocessing. Multithreading refers to running multiple threads within a single process, where threads share the same memory space. Multiprocessing, on the other hand, involves multiple independent processes, each with its own memory space. These processes can run on separate CPU cores, allowing for better utilization of hardware resources.

2.3.3 Types of Parallelism

Parallelism in programming can generally be divided into two broad categories: data parallelism and task parallelism:

- **Data Parallelism** - involves distributing data across multiple processing units where each unit performs the same operation on different chunks of the data. This type of parallelism is especially effective in scenarios where large datasets need to be processed, such as in image processing, matrix operations, or large-scale simulations;
- **Task Parallelism** - refers to the execution of different tasks or operations concurrently. Unlike data parallelism, where the same operation is applied across various data elements, task parallelism assigns different operations or parts of a problem to different processing units.

Figure 2.9 illustrates Data and Task parallelisms. Both types of parallelism can be combined in hybrid systems, depending on the complexity and structure of the problem at hand.

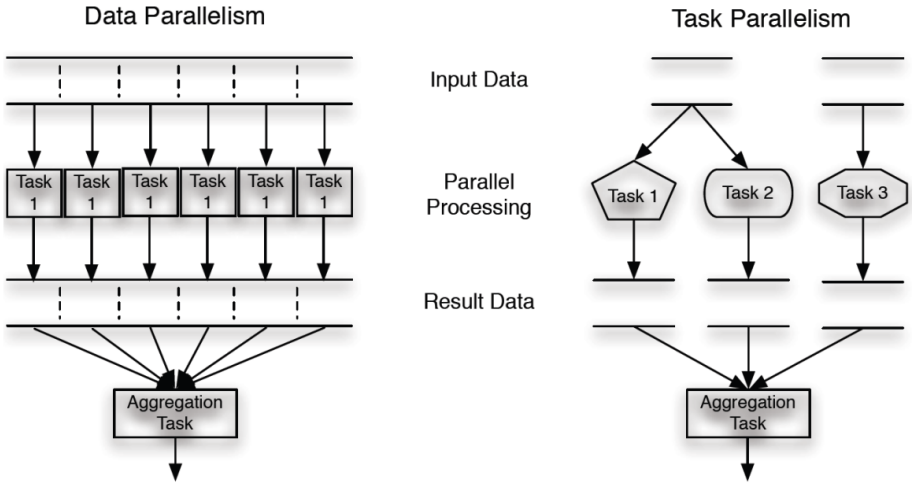


Figure 2.9: Types of parallelism

The choice between data parallelism and task parallelism depends on the nature of the workload and the architecture of the system. One common application of parallelism is in scientific simulations like fluid dynamics, climate modeling, and astrophysics, where vast amounts of data must be processed simultaneously to simulate physical phenomena [Gropp and Smith, 1990, Jacob et al., 2012, Schaller et al., 2024], in such simulations, the dataset can be partitioned across multiple processing units, with each unit responsible for processing a different region of the data space. Similarly, neural networks and machine learning models rely heavily on data parallelism to train models on massive datasets, particularly when leveraging GPUs to process large matrices [Li et al., 2020].

2.3.4 Performance Measurement

Evaluating the performance of parallel programs is essential to understand how well the parallelization strategy scales and utilizes hardware resources. Two widely used metrics are *speedup* and Amdahl's Law speedup, both of which compare the parallel program's execution time to that of its sequential counterpart.

Speedup. Speedup (S) is defined as the ratio of the time it takes to execute a task on one processor ($T_{sequential}$) to the time it takes on n processors ($T_{parallel}$):

$$S = \frac{T_{sequential}}{T_{parallel}} \tag{2.2}$$

This gives insight into how much faster a program runs when parallelized across multiple processors. In an ideal scenario, the speedup increases linearly with the number of processors, i.e. doubling the processors would halve the runtime. However, due to overhead by the Operating System or communication between processors, speedup is often sub-linear in practice.

Amdahl's Law. Amdahl's Law [Amdahl, 1967] provides a theoretical limit to the speedup that can be achieved by parallelizing a program, based on the proportion of the program that can be parallelized. If P represents the fraction of a program that can be parallelized, then the remaining $1 - P$ must still be executed sequentially. The maximum achievable speedup, as per Amdahl's Law, is given by:

$$S_{max} = \frac{1}{(1 - P) + \frac{P}{N}} \tag{2.3}$$

As N (the number of processors) increases, the term $\frac{P}{N}$ shrinks, but the sequential part $1 - P$ remains a bottleneck, limiting the overall speedup. This law highlights that even with infinite processors, speedup is limited by the portion of the program that is inherently sequential.

3

Frameworks

The systems supporting this work are essential for analysing and correcting the geolocation errors in GEDI data. This chapter delves into the design and functionality of the two primary frameworks employed: GEDI Simulator and GEDICorrect. First, the GEDI Simulator [Hancock et al., 2019], an existing tool, is explored, with its standard geolocation correction methods analyzed to establish a performance benchmark for the newly introduced framework. GEDICorrect, in turn, brings novel footprint-level correction methods and incorporates parallel processing to enhance performance and scalability.

Understanding the foundations of both frameworks is crucial for their application in the experiments described in Section 4.8. By grasping how these systems operate, it becomes evident why they are central to improving the accuracy of GEDI data, particularly in areas where geolocation errors significantly impact canopy height and biomass estimations.

3.1 GEDI Simulator

The GEDI Simulator[1] is a set of programs designed to simulate GEDI-like waveform data [Hancock et al., 2019]. It generates vegetation and terrain metrics by leveraging small-footprint datasets, such as ALS point cloud data, which consists of individual laser returns from vegetation and terrain. ALS data is often characterized by high spatial resolution and provides precise 3D point clouds of the surveyed area. This section introduces the tool and its separate programs for simulation and metrics extraction.

3.1.1 Overview

The GEDI Simulator is designed to replicate the waveform generation process of the GEDI mission using the method proposed by Blair and Hofton [Blair and Hofton, 1999]. The simulator mimics the GEDI instrument by generating a full-waveform from the discrete-returns ALS data. To do that, the simulator aggregates individual ALS returns within a large circular footprint, similar in size to the GEDI footprint (25-meter diameter). These discrete returns are then "binned" along the vertical axis to create a continuous waveform, which represents the distribution of energy reflected by the vegetation and terrain surfaces within the footprint. Each waveform bin, corresponds to a specific height above the ground and the relative energy in each bin, representing the density of the scatterers (leaves, branches, or ground) at that height.

To produce a GEDI-like waveform, the simulator incorporates key characteristics of the actual GEDI instrument, such as i) adding the instrument's noise to the simulated waveforms to better reflect real-world observations; and ii) calculating GEDI-specific Relative Height (RH) metrics (e.g. RH100, RH95, RH75, etc), which represent the height below which a certain percentage of waveform energy is returned. These metrics are crucial for understanding the vegetation vertical structure. Besides the RH profile, the simulator also generates terrain elevation information by analyzing the waveform's return from the ground, where the waveform's significant last return usually corresponds to the ground level. A key feature of the simulator is the `collocateWaves` program, which aligns or "collocates" the reported GEDI waveform with the ALS simulated waveform using a correlation method introduced in Blair and Hofton [Blair and Hofton, 1999]. This ensures that GEDI-reported waveform accurately reflects the vegetation structure and topography captured in the ALS data.

In summary, the simulation technique involves translating the discrete-return ALS point cloud data, acquired in a set of footprints, into a waveform that mimics the resolution and FWHM (for GEDI, FWHM = 15 ns) of a GEDI footprint [Hancock et al., 2019]. This simulation will generate the RH metrics (from RH0 to RH100) and terrain data, which can be used to calculate key parameters such as canopy height, vegetation density and terrain elevation.

The following section focuses on the internal mechanisms of the framework, organized into distinct programs. Each program handles a specific step in the overall simulation process. For instance, `gediRat` simulates GEDI-like waveforms using ALS data, `gediMetrics` extracts metrics and RH profiles from either reported GEDI or simulated GEDI footprints, and finally `collocateWaves` aligns GEDI-reported waveform with the simulated waveform derived from the ALS data. This alignment is an attempt to correct the GEDI footprint geolocation error, which is estimated to be around 10.3 meters in GEDI Version 2 data.

[1] https://bitbucket.org/StevenHancock/gedisimulator/src/master/

3.1.2 GEDI Simulator Programs

The GEDI Simulator is composed of three key programs, which are detailed below.

gediRat

The `gediRat` program is responsible for simulating GEDI-like waveforms from ALS data (`.las` files) by converting the dense point cloud into a continuous waveform representation. The program allows the user to specify the locations of GEDI footprints by introducing Latitude and Longitude coordinates in the same CRS as the input ALS data. It outputs the simulated waveforms in either HDF5 or ASCII format. The HDF5 file follows the same file structure and hierarchy as GEDI L1B data products.

Some of the most important command options for `gediRat` include:

- `-inList` - Specifies input file in ASCII format of a list of the absolute paths to the `.las` files
- `-hdf` - Declares that the output file will be in HDF5 format;
- `-output` - Specifies the output filename. If the `-hdf` command is selected, output filename must have the `.h5` extension;
- `-ground` - Includes the ground portion of the waveform in the output;
- `-aEPSG` - Defines the target EPSG for the input ALS data;
- `-coord $lon $lat` - Simulates a waveform at a single set of Longitude and Latitude coordinates. These coordinates must be in the same CRS as the input ALS;
- `-maxBins` - Describes the maximum number of height bins (Z) in which to make the simulation. Defaults to 1024 bins;
- `-listCoord` - Specifies a list of Longitude and Latitude coordinates in ASCII format file.

In addition, the user can specify instrument parameters such as Pulse Width, FWHM, Footprint Width, though these default to the GEDI instrument's characteristics.

The output of `gediRat` is an HDF5 file with the simulated waveforms at specified coordinates. An HDF5 file is a hierarchical data format[2] used to store large amounts of structured data, allowing for efficient storage and access of complex datasets. It organizes data into groups and datasets, enabling flexible management of diverse data types, including multi-dimensional arrays and metadata, making it well-suited for handling the large volumes of waveform data produced by GEDI simulations.

After simulating the GEDI L1B waveforms product for the selected footprints, the output of `gediRat` can be read in the `gediMetrics` program for extracting relevant metrics such as the RH profile, ground height, and other structural information, which are essential for further analysis and comparison with GEDI data products (e.g. L1B and L2A).

It is worth noting that the simulated waveforms do not have noise (unlike real GEDI footprint waveforms, which have small fluctuations in waveform amplitude) and are extended to the default maximum of 1024 height bins. This number can be adjusted by providing a number of bins to the `-maxBins` command.

[2] https://www.hdfgroup.org/

gediMetrics

`gediMetrics` processes real or simulated GEDI L1B data to produce standard waveform metrics. These metrics include ground slope and elevation, canopy cover, leading and trailing edge extents, RH profile metrics using three algorithms (Gaussian, Inflection and Maximum), FHD and Leaf Area Index (LAI) [Hancock et al., 2019, Beck et al., 2021]. The program also generates metrics that are unavailable in official GEDI data products, such as ground elevation from ALS, ground slope, ALS cover, RH profile using the true ground from the ALS data, ALS point density and ALS beam density within each footprint. This process mimics the transformation of GEDI L1B to L2A and L2B data products.

When processing each waveform signal from the input, the data is first denoised by removing points above a threshold of $mean+5\sigma$ and smoothed with 0.75 times the pulse width (p_σ) [Hancock et al., 2019]. After this preprocessing, there are three main methods for processing the signal:

- **Gaussian Fitting** - This method fits the waveform to Gaussian curves using the Levenberg-Marquardt optimisation [Levenberg, 1944]. It selects ground and vegetation returns by calculating the percentage of waveform energy. The resulting variables include "*rhGauss*", "*gHeight*" and "*gaussHalfCov*";
- **Maximum** - This method identifies the lowest maximum point in the waveform and assumes it as the ground return, using the raw waveform without further fitting. The resulting variables include "*rhMax*", "*maxGround*" and "*maxHalfCov*";
- **Inflection Points** - This method identifies the lowest two inflection points in the waveform. The ground return is then determined by calculating the center of gravity between these points. The resulting variables include "*rhInfl*", "*inflGround*" and "*inflHalfCov*".

Some crucial command options for `gediMetrics` include:

- `-input` - Specifies input waveform file from either gediRat or reported GEDI L1B data;
- `-readHDFgedi` - Informs the program that the input file is in HDF5 format;
- `-outRoot` - Specifies output filename root string. The output will always be a "metric.txt" file;
- `-ground` - Informs the program that the ground should be read from the input file;
- `-varScale` - Selects variable noise threshold scale (multiple of standard deviation above mean to set threshold);
- `-sWidth` - Selects smoothing width after denoising the waveform;
- `-rhRes` - Selects the percentage energy resolution of the RH Profile;
- `-laiRes` - Selects the vertical resolution of the LAI profile in meters. It defaults to 10m.

collocateWaves

The `collocateWaves` program is responsible for finding an optimal placement of GEDI footprints by aligning them with ALS data, thus correcting for geolocation errors. The goal is to find an optimal (X, Y, Z) placement vector that maximizes the correlation between the reported and simulated GEDI waveforms for the entire orbit of the input file. For this, the simulator describes three modes of operation:

- **Bullseye** - This mode tests a grid of (X, Y, Z) transformations and calculates the correlation between points and finds the highest correlated simulated point, similar to the one described in Blair and Hofton [Blair and Hofton, 1999], where the best fit is found through exhaustive searching on a grid of possible transformations.
- **Simplex** - The simplex algorithm is an optimization technique to find the optimal (X, Y, Z) along an error surface. Starting with an initial guess for the footprint placement (which should be within approximately 20 meters of the true location), it adjusts the placement iteratively. The algorithm then explores the error surface and shifts the position in the direction where the correlation improves, stopping when no other optimal placement is found.
- **Annealling** - This is a hybrid method that combines both the **Bullseye** and **Simplex** approaches. First, a grid search is performed using the Bullseye method until the best candidate is found. Then, the Simplex algorithm is employed from the best position of the grid to refine the placement and to achieve a higher level of precision in the alignment.

Figure 3.1 illustrates these modes of operation, with the green circle representing the reported GEDI footprint location, while the blue circle refers to the potential best geolocation of the reported footprint inside the figure.

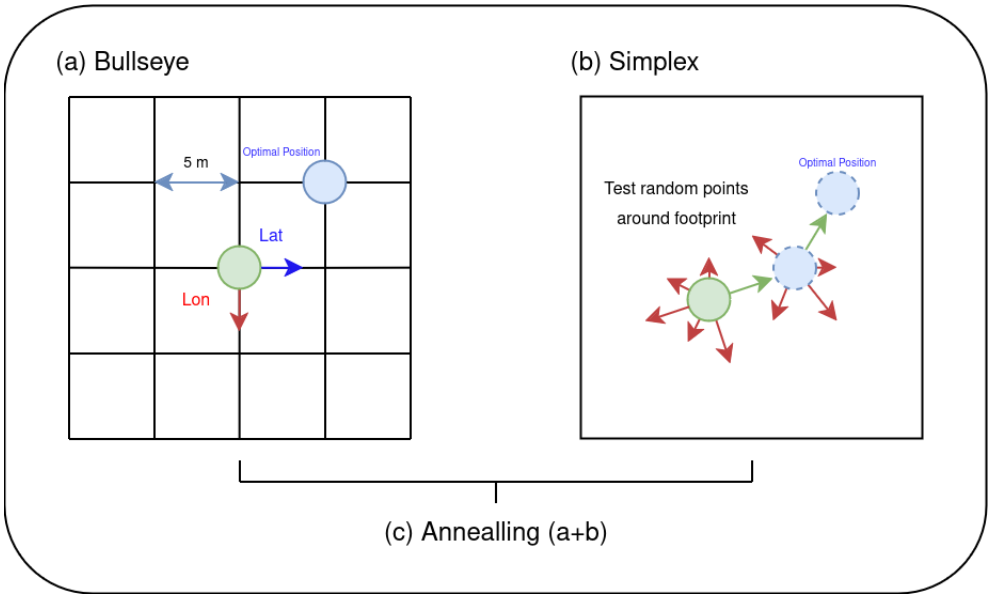


Figure 3.1: Illustration of `collocateWaves` Bullseye (a), Simplex (b) and Annealling (c) modes of operation used in GEDI Simulator.

In all modes, the comparison between reported and simulated GEDI waveforms is performed using the Pearson Correlation. Overall, the program stores each footprint's correlation (ρ) in an array and calculates the mean correlation of the entire orbit:

- For the *bullseye* mode, it selects the highest correlated simulated point from the initial position and saves the affine transformation with the correlation value.
- In *simplex* mode, this mean correlation is returned by a function where the `gsl_multimin_fminimizer` intervenes. This function allocates a Minimizer[3] procedure that

[3] https://www.gnu.org/software/gsl/doc/html/multimin.html

minimizes the error surface expressed as $(1 - \rho)$, resulting in the optimal affine transformation with the highest mean correlation for the orbit.

After finding the optimal simulated points, the program outputs two files: the corrected position footprints in HDF5 format (structured similarly to the output of `gediRat`), and a Text file that contains details about the offset used and the mean correlation of the orbit. The HDF5 file can also be introduced as input to `gediMetrics`, producing their respective simulated waveform metrics.

The most crucial command options to run `collocateWaves` are as follows:

- `-listAls` - Specifies an ASCII file containing the absolute paths to the ALS data files;
- `-listGedi` - Specifies an ASCII file containing the absolute paths to the input GEDI L1B data product files that need correction;
- `-readHDFgedi` - Informs the program to read the GEDI files in HDF5 format;
- `-aEPSG` - Sets the EPSG code for the ALS data;
- `-solveCofG` - Defines a center of gravity to match vertical offsets (matches ALS ground discrepancy with GEDI data);
- `-geoError $expError $correlDist` - Adjusts rapids geolocation by defining the expected geolocation error and the correlation distance;
- `-minDense` - Specifies minimum number of ALS beams/m^2 to create simulations;
- `-minSense` - Selects footprints with specified minimum sensitivity;
- `-writeWaves $outFile` - Outputs corrected waveforms in HDF5 format with specified output filename;
- `-simplex` - Uses the *Simplex* algorithm;
- `-anneal` - Uses the *Annealling* algorithm.

3.1.3 Performance of Geolocation Correction in `collocateWaves`

The geolocation correction algorithm, as implemented in `collocateWaves`, has proven to be highly inefficient. More specifically, initial tests using the *Simplex* mode required ≈90 hours to complete, and resulted in an R^2 of 0.52 when comparing the reported RH95 values with the simulated RH95 (see Section 5.1). This level of performance is far from ideal, considering the amount of time and resources required for such modest correlation.

Despite the framework's functionality, the initial tests revealed clear limitations in both performance and ease of use. Several improvements could be made to address these issues:

1. **Parallel Execution** - Currently, parallel processing is only possible by executing multiple bash commands (using the `&` command) to run the programs. This requires multiple setup steps, such as dividing the desired study area into smaller chunks, which complicates the user experience and reduces research efficiency. Therefore, a more parallelized implementation of the core algorithms could

significantly reduce its runtime. Specifically, optimizing the codebase for multi-threading or GPU-based processing would speed up operations without compromising accuracy. This would involve adapting the framework to incorporate modern solutions, such as using locks, atomic operations, or CUDA for parallel execution.

2. **Memory Allocation** - The performed tests consumed ≈35GB of RAM due to loading all input GEDI files and ALS data into memory at once. Executing N multiple parallel processes under this approach would result in $35 \times N$ GB of RAM allocation. This excessive resource demand limits the program's usability on most standard personal computers. To address this, memory allocation issues should be identified and optimized, and memory management techniques, such as freeing unused memory should be integrated.

3. **New methods for GEDI geolocation correction** - Currently, the GEDI geolocation error correction, performed within the `collocateWaves` program, relies solely on the waveform matching method using Pearson's correlation as the metric. The correlation between GEDI-reported waveform and simulated waveform is used to determine the best location of GEDI measurements, i.e., the location where the reported and simulated waveforms show the highest correlation. Introducing additional methods, such as terrain matching and RH profile matching, along with different criteria (e.g. correlation-based, distance-based, and divergence-based) and metrics (e.g. Pearson, Spearman, divergence index), could improve the alignment of GEDI-reported waveform with the simulated waveform. This would result in a more accurate GEDI geolocation correction.

Moreover, correcting GEDI footprint geolocation at the orbit level is not ideal for study areas with high heterogeneity. As pointed out in Section 1.1 of this dissertation, the core assumption of the *collocateWaves* program is that there is a constant systematic offset along the orbit, which is likely unrealistic due to the high-frequency errors associated with GEDI [Tang et al., 2023]. A footprint-level approach to correcting geolocation errors, which assumes that each footprint has its own unique offset rather than a uniform error across the orbit, would better account for local variability and improve geolocation accuracy.

The described performance issues, along with the program's lack of flexibility in memory management, parallelization, and code structure, highlight the need for a more advanced approach. The newly proposed framework, *GEDICorrect*, addresses these shortcomings by providing a more efficient and user-friendly solution.

3.2 GEDICorrect

The proposed framework, *GEDICorrect*, is developed to address the limitations identified in the GEDI Simulator (see Section 3.1.3). While GEDI Simulator uses an orbit-level correction approach, *GEDICorrect* introduces a footprint-level correction method which is particularly beneficial in areas with significant horizontal variability in tree height and cover. Figure 3.2 shows a visual representation of the GEDI footprint geolocation correction, where the green-colored footprints are the reported GEDI footprints and the blue-colored are the optimal geolocation for each reported footprint. While `collocateWaves` shifts the entire orbit, GEDICorrect's correction method consists of simulating random points around each reported footprint (Figure 3.2(a)). To accelerate waveform simulation and metrics calculation, the framework supports parallel processing and is designed to be scalable, making it a faster and more flexible solution than the GEDI Simulator.

More specifically, *GEDICorrect* was designed to:

1. **Parallelization**: The framework supports both sequential and parallel execution of the geolocation

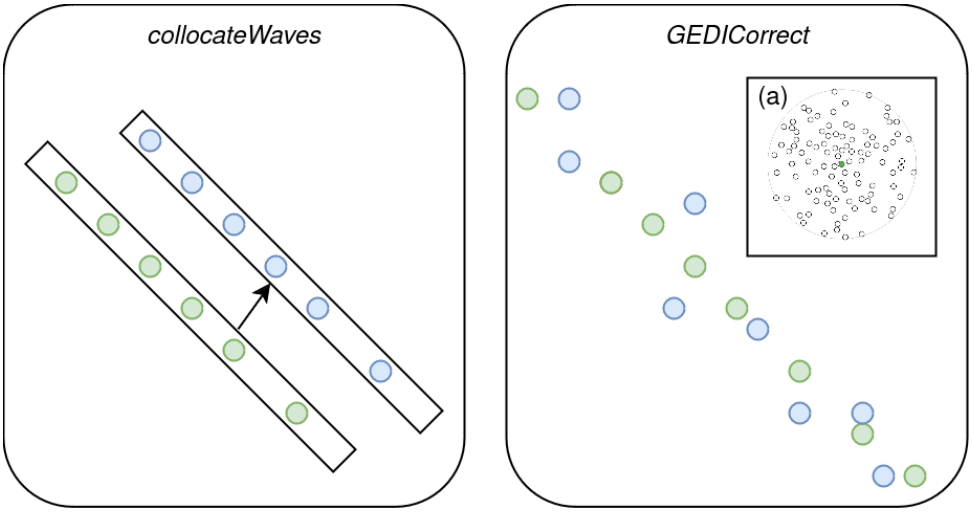


Figure 3.2: General overview of the footprint geolocation correction method of both frameworks.

correction process, ensuring scalability for large datasets;

2. **Efficient Memory Management**: It employs efficient memory practices, such as utilizing structured data formats (e.g., `pandas` DataFrames) and limiting read/write operations to when necessary;

3. **New methods for GEDI geolocation correction** - It introduces new methods for footprint geolocation correction, such as Terrain Matching and RH Profile Matching, providing improved accuracy.

3.2.1 Framework Design

The framework is composed of 4 main units: i) input; ii) simulation; iii) scoring; and iv) output (see Figure 3.3). The simulation unit is mostly handled by subprocesses that call `gediRat` and `gediMetrics`. The sections below describe the design of *GEDICorrect* in further detail, focusing on the improvements made to enhance the framework's performance over existing methods.

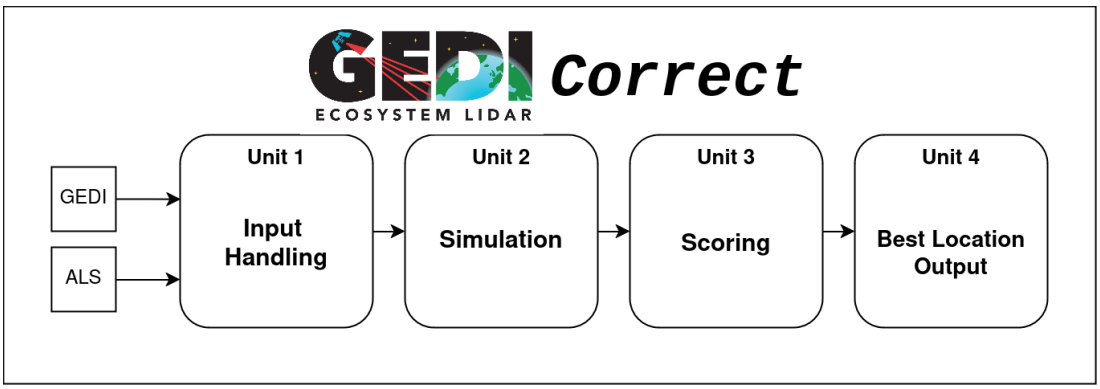


Figure 3.3: The GEDICorrect Framework

Unit 1: Input

The input to *GEDICorrect* consists of the merged GEDI L1B and L2A data products along with a directory containing the corresponding ALS (`.las`) files. To ensure proper functionality, the framework performs an initial sanity check on all input files, which will be described below.

First, each `.las` file is read and parsed, and a bounding dictionary is created with the structure presented in the following example:

```
{'ALS000001.las': Polygon(...),
 'ALS000002.las': Polygon(...),
                  ... }
```

Two methods are available for generating these bounds:

- **Simple Bounding Box** - This method is the fastest and uses the `minX`, `minY`, `maxX`, and `maxY` coordinates from the ALS header to create a simple rectangular boundary around the point cloud data;
- **Convex Hull Algorithm** - Generates a convex hull [Andrew, 1979] around the point data to create a tight-fitting boundary, which ensures a better spatial representation of the ALS data. This method is both more accurate and time-consuming.

If this is GEDICorrect's first run with the ALS data, it will save the bounds in Shapefile (`.SHP`) format, which will be used in subsequent runs. This unit marks the beginning of **Efficient Memory Management**, addressing a key limitation of the GEDI Simulator, which reads the entire ALS dataset into memory during each run. By storing the bounds, *GEDICorrect* optimizes memory usage for future processes. Once the ALS bounds are created, the framework loads all the GEDI files by reading them from *Geopackage* (`.GPKG`) files, which is a compressed format of a dataset containing geospatial information[4]. For each footprint in each GEDI dataset, a square buffer is generated around the centroid to identify intersections with the ALS data. In this project, each square buffer is set to 50 meters, which ensures the size of two whole footprints (which have 25 meters in diameter).

Any GEDI footprints and their corresponding buffers that are not entirely inside the ALS bounds are discarded to ensure the correction process focuses only on valid areas where simulation is possible (see Figure 3.4). If any files are corrupted or missing, the sanity check will fail, prompting the user to provide a new set of inputs.

After the sanity check is completed, the output of this unit is a validated list, containing only footprints within the ALS bounds, ready for further processing in the next unit. Figure 3.5 illustrates the pipeline of this entire unit in detail.

[4] https://www.geopackage.org/

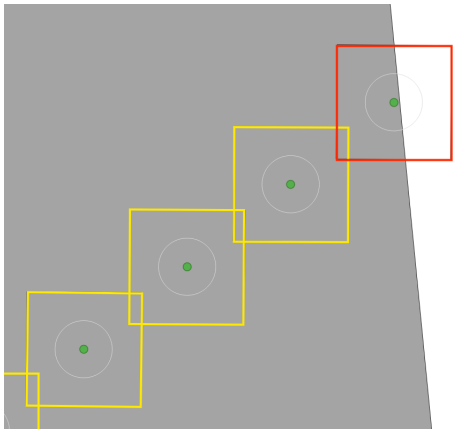


Figure 3.4: GEDI footprint inside ALS bounds verification process. Any footprint and respective 50 meter buffer that is not within ALS bounds is discarded for correction.

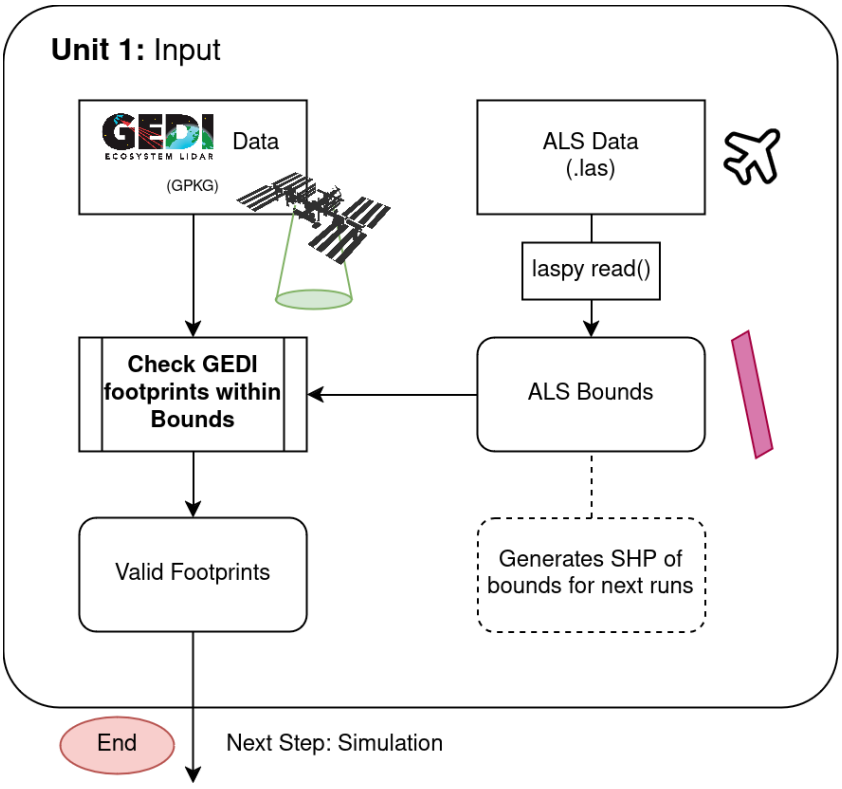


Figure 3.5: Input Unit of the GEDICorrect Pipeline where the input GEDI files and ALS data are passed through a sanity check.

Unit 2: Simulation

For each verified GEDI footprint, a random set of N points is generated within a specified radius around the footprint centroid, with a maximum distance of 12.5 meters from the reported centroid and a minimum distance of 1 meter between generated points. These parameters, however, are configurable based on the study area or user preferences.

Once the points are generated, their respective latitude and longitude coordinates are saved in an ASCII file,

which serves as input to the `gediRat` program. The `gediRat` program simulates waveforms from ALS point cloud for each of the generated points, producing an HDF5 file (`simulated.h5`). Next, the `gediMetrics` program is executed to produce waveform metrics and RH profiles for each of the N generated points, generating a Text file (`metrics.txt`). Both programs, `gediRat` and `gediMetrics`, are run using the `subprocess` Python library.

Both output files, HDF5 and TXT, are parsed into `pandas` DataFrames and then concatenated into a unified dataset. A DataFrame is a two-dimensional, tabular data structure commonly used in data analysis, where data is arranged in rows and columns, similar to a spreadsheet. Each column in a DataFrame can hold different data types, and it provides flexible indexing[5]. To ensure proper geospatial representation between subsequent operations, the concatenated DataFrame is transformed into a `geopandas` GeoDataFrame by adding geometry in the form of latitude and longitude coordinates for each generated point.

Additionally, a filtering task is applied to account for time differences between the GEDI and ALS data acquisition. If the vertical offset between the two datasets (Reported GEDI and Simulated GEDI) exceeds a threshold of 10 meters, it is flagged as a discrepancy, potentially indicating vegetation changes over time between the GEDI and ALS data acquisition or inaccurate simulation results.

The final output from this simulation unit is a filtered and concatenated GeoDataFrame, which contains the simulated points and their associated metrics. This dataset is appended to a final simulation list, where all of the simulated footprints are located. When this Simulation Unit finishes processing, the result list will be passed onto the Scorer Unit. Figure 3.6 illustrates this unit in detail.

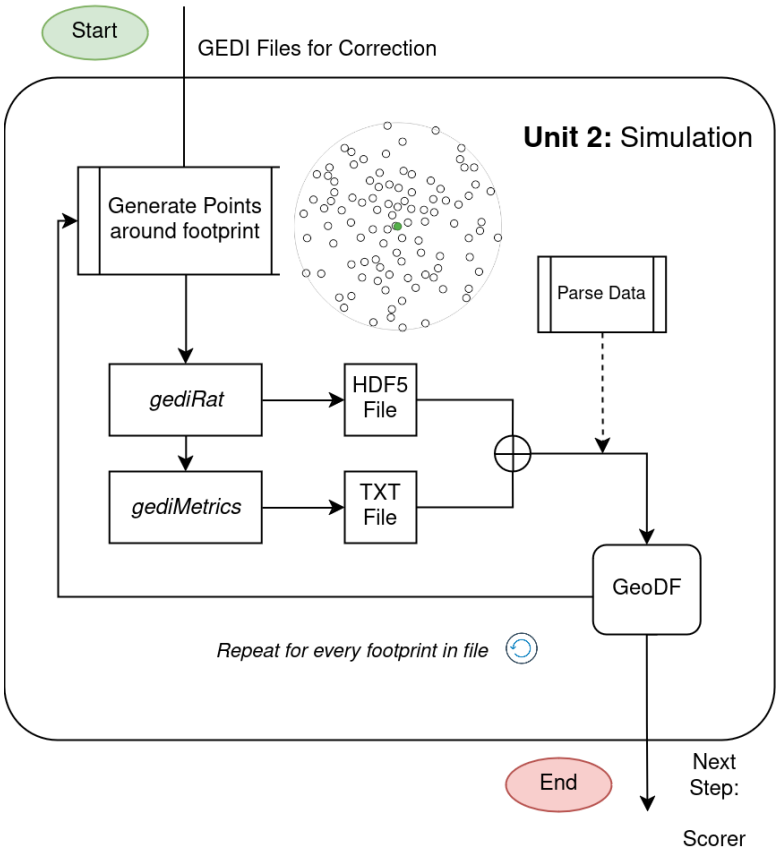


Figure 3.6: Simulation Unit of the GEDICorrect Pipeline.

[5] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Unit 3: Scoring

While the Waveform Matching method implemented in the GEDI simulator relies solely on Pearson's correlation analysis between the reported and simulated waveform energy values, GEDICorrect introduces two additional methods: Terrain matching and RH profile matching. Figure 3.7 illustrates the methods. These additions represent one of the major improvements over GEDI Simulator: **New Methods, Criteria and Metrics**.

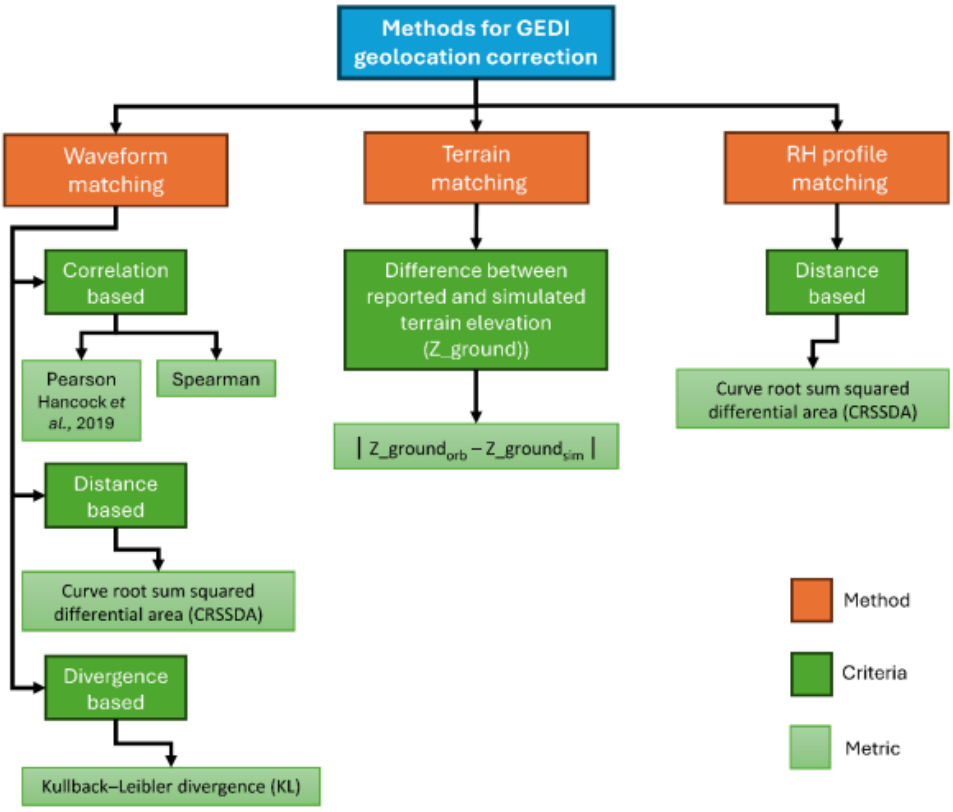


Figure 3.7: Flowchart illustrating the methods, criteria, and metrics tested for GEDI geolocation correction.

Waveform Matching. In addition to Pearson's correlation, the Spearman's Correlation formula was also implemented for waveform matching, which is more suitable for non-linear data, such as GEDI waveforms.

For the distance-based criteria between waveform energies, the Curve Root Sum Squared Differential Area (CRSSDA) [Zhou et al., 2016] metric was implemented, an area-based measure that assesses curve similarity by calculating the area between a reported (r) and a simulated (s) waveforms. The method first determines the squared difference between the two waveform curves at each height bin (Z), sums these differences across all bins, and takes the square root to obtain the waveform curve similarity between the start and end (n) locations of the waveform. Equation 3.1 describes how to calculate CRSSDA. A smaller value of CRSSDA indicates a higher curve similarity between two waveforms.

$$CRSSDA = \sqrt{\sum_{i=0}^{n}(r_i - s_i)^2} \tag{3.1}$$

For the divergence-based criteria, the Kullback–Leibler (KL) metric was used. KL measures the similarity between two probability distribution functions [Kullback and Leibler, 1951]. KL has been successfully applied in fields such as image pattern recognition, hyperspectral image classification, and waveform matching [Olszewski, 2012, Zhou et al., 2016]. Since a waveform can be normalized as a probability distribution function, the KL divergence metric was used to assess the similarity between the reported (r) and the simulated (s) waveform using Equation 3.2. A smaller value of KL indicates a higher curve similarity.

$$KL = \sum_{i=0}^{n} log(r_i/s_i) * r_i \tag{3.2}$$

Terrain Matching. For the terrain evaluation, the difference-based criteria involves matching the ground elevation from the reported GEDI to the ALS simulated ground elevation (Equation 3.3). The absolute value of the smallest elevation difference is granted a higher score.

$$Z\_DIFF = |Z\_Ground_r - Z\_Ground_s| \tag{3.3}$$

RH Profile Matching. Regarding the RH profile matching method, an adapted CRSSDA equation from 3.1 was implemented to create a distance-based metric between the reported and simulated RH values at different intervals. The evaluation begins at RH25, increasing by 5% increments, up to RH100, capturing the whole RH profile of the vegetation and its internal structure. The adapted CRSSDA metric is described in Equation 3.4, where r_{RH_i} represents the reported RH value at interval i, and s_{RH_i} represents the simulated RH value at the same interval.

$$CRSSDA\_RH = \sqrt{\sum_{i=0}^{n}(r_{RH_i} - s_{RH_i})^2} \tag{3.4}$$

Essentially, this metric measures the area between the reported and simulated RH profile curves (see Figure 3.8), with a CRSSDA_RH value close to zero indicating perfect alignment between the reported and simulated waveforms.

Once all input footprints have been simulated, the data is passed to the *Scorer* class, where it identifies the optimal simulated footprint that best aligns with the corresponding reported GEDI footprint. Figure 3.9 illustrates the process.

The *Scorer* class, contains all of the required functions to calculate the previously described metrics. The available criteria are represented as a list of strings within the class, allowing the user to select the desired metric, or a combination of metrics:

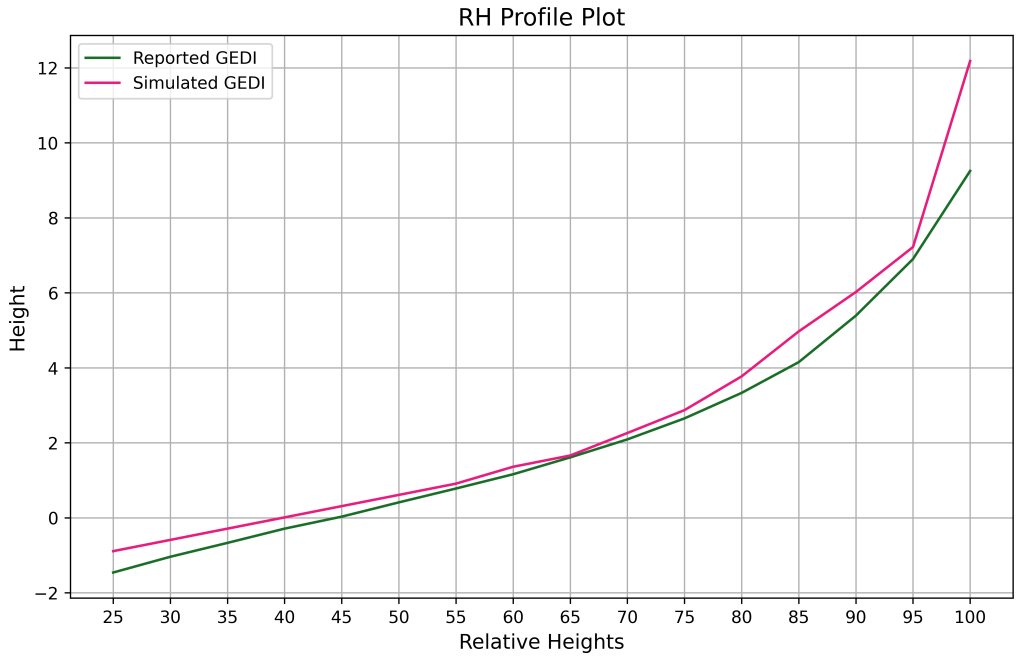


Figure 3.8: Example plot of the entire RH Profile. The adapted CRSSDA is calculated for the entire RH profile at intervals of 5%, from 25% to 100%.

- `'wave'` - Spearman's correlation;
- `'wave_distance'` - Distance based CRSSDA for the waveforms;
- `'kl'` - Divergence based KL metric for the waveforms;
- `'terrain'` - Difference between reported and simulated terrain elevation;
- `'rh_distance'` - Distance based CRSSDA for the entire RH profile.

To select multiple metrics, the user can provide an option by adding a '+' sign between desired metrics. Each metric has an individual score ranging from 0 to 1, where 1 represents the highest score (indicating greater similarity with the reported footprint) that a simulated footprint can achieve. After computing the scores for all selected criteria, the *Scorer* class updates the GeoDataFrame by adding a column for each metric, along with a *final_score* column. The final score is calculated by summing all of the selected individual metric scores and dividing by the number of selected metrics, as shown in Equation 3.5, where $metric_i$ is an individual metric score, and $n\_criteria$ is the total number of selected metrics for scoring. Since each metric score ranges from 0 to 1, the final score will be ranging from 0 to 1, with 1 being the highest score a simulated footprint can obtain.

$$final\_score = \frac{\sum metric_i}{n\_criteria} \tag{3.5}$$

In summary, both Unit 2 (Simulation) and Unit 3 (Scoring) were designed to be executed sequentially or in parallel, which represents a clear improvement over GEDI Simulator by enhancing the geolocation correction process through parallelization. The methods for parallel implementation are described in Section 4.5.1.

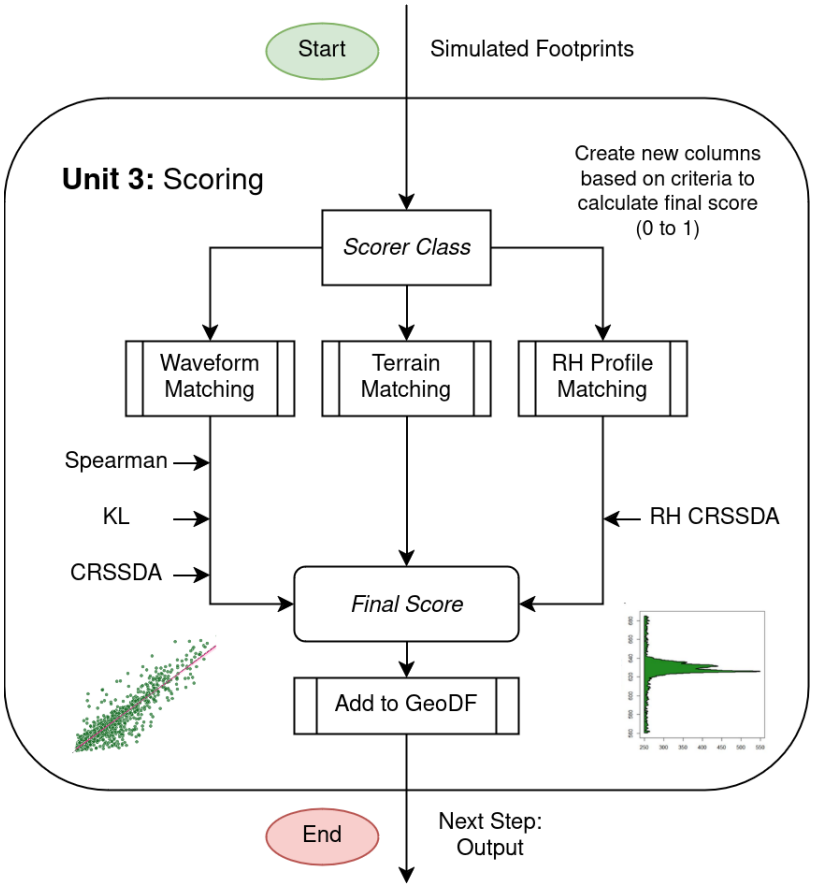


Figure 3.9: Scoring Unit of the GEDICorrect Pipeline.

Unit 4: Output

Programatically, the output of the *Scorer* class from Unit 3 consists of a list of GeoDataFrames containing both the simulated footprints and their respective scores (scores ranging from 0 to 1, for each metric and final score). The pipeline offers the option to save the scored simulated footprints in three operational modes: i) all simulated points generated around each reported footprint; ii) the highest-scored simulated footprints; and iii) simulations at the original locations of the reported footprints. Figure 3.10 illustrates these output operating modes.

The selection of the best location for each reported footprint is performed by sorting the GeoDataFrame regarding the *final_score* column (previously added during the Scoring Unit) and selecting the simulated footprint with the highest final score. The selected best footprint (with the highest score), which is expected to represent the "true" geolocation of the reported GEDI measurements, is chosen for output (see Figure 3.11). The final output can be saved in both *Shapefile* (`.SHP`) and *Geopackage* (`.GPKG`) formats, enabling easy visualization in spatial analysis GIS tools such as QGIS.

If multiple GEDI files were introduced during Unit 1, the framework repeats the geolocation correction process from Unit 2 (Simulation) for each subsequent GEDI file, repeating Units 3 and 4, until all files have been processed.

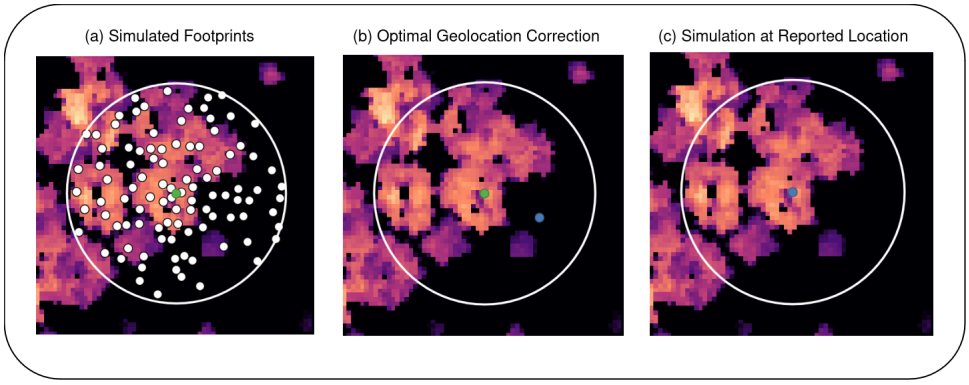
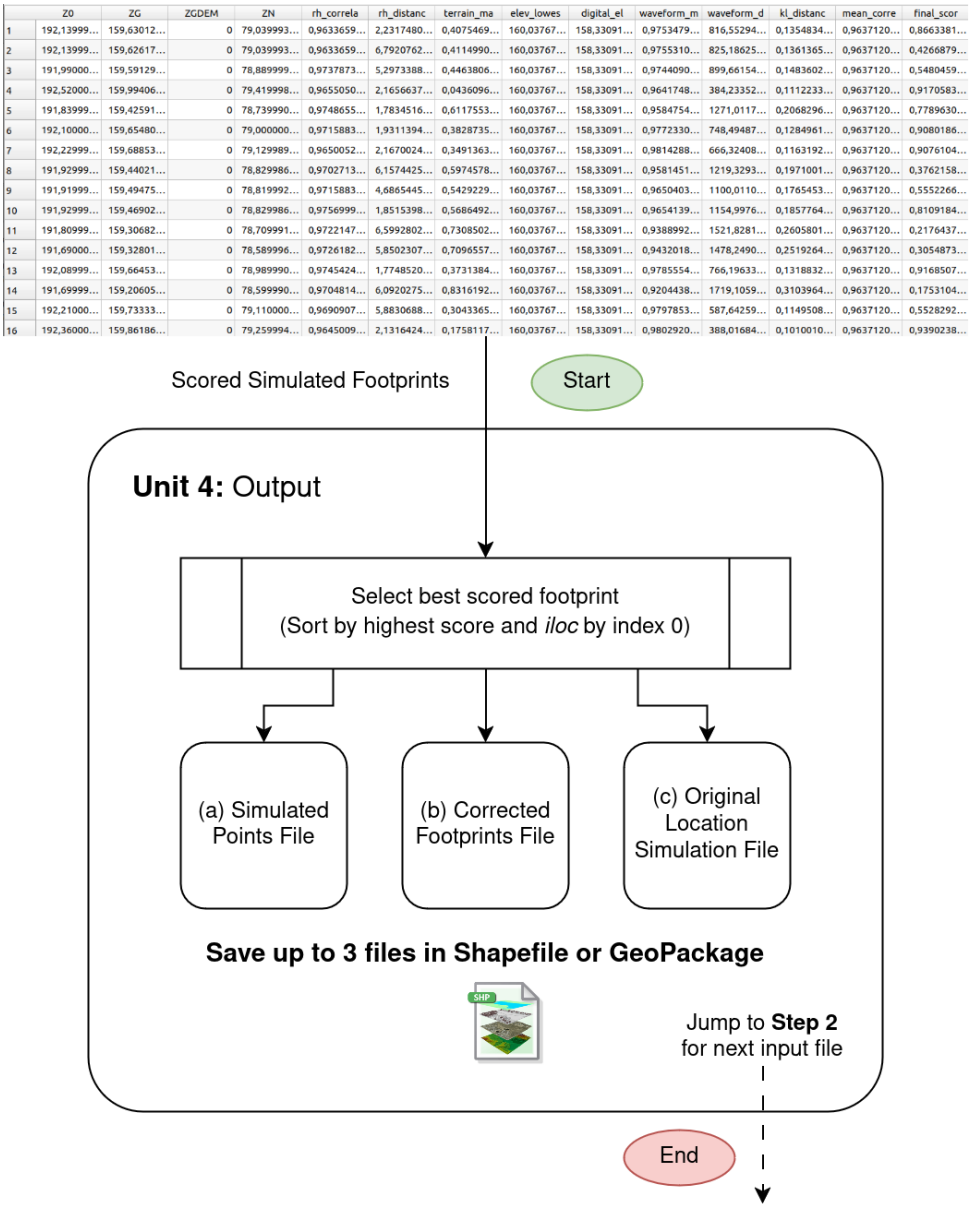


Figure 3.10: Saving modes in Unit 4 of the GEDICorrect Pipeline. Files are either saved to Shapefile or GeoPackage.



	Z0	ZG	ZGDEM	ZN	rh_correla	rh_distanc	terrain_ma	elev_lowes	digital_el	waveform_m	waveform_d	kl_distanc	mean_corre	final_scor
1	192,13999...	159,63012...	0	79,039993...	0,9633659...	2,2317480...	0,4075469...	160,03767...	158,33091...	0,9753479...	816,55294...	0,1354834...	0,9637120...	0,8663381...
2	192,13999...	159,62617...	0	79,039993...	0,9633659...	6,7920762...	0,4114990...	160,03767...	158,33091...	0,9755310...	825,18625...	0,1361365...	0,9637120...	0,4266879...
3	191,99000...	159,59129...	0	78,889999...	0,9737873...	5,2973388...	0,4463806...	160,03767...	158,33091...	0,9744090...	899,66154...	0,1483602...	0,9637120...	0,5480459...
4	192,52000...	159,99406...	0	79,419998...	0,9655050...	2,1656637...	0,0436096...	160,03767...	158,33091...	0,9641748...	384,23352...	0,1112233...	0,9637120...	0,9170583...
5	191,83999...	159,42591...	0	78,739990...	0,9748655...	1,7834516...	0,6117553...	160,03767...	158,33091...	0,9584754...	1271,0117...	0,2068296...	0,9637120...	0,7789630...
6	192,10000...	159,65480...	0	79,000000...	0,9715883...	1,9311394...	0,3828735...	160,03767...	158,33091...	0,9772330...	748,49487...	0,1284961...	0,9637120...	0,9080186...
7	192,22999...	159,68853...	0	79,129989...	0,9650052...	2,1670024...	0,3491363...	160,03767...	158,33091...	0,9814288...	666,32408...	0,1163192...	0,9637120...	0,9076104...
8	191,92999...	159,44021...	0	78,829986...	0,9702713...	6,1574425...	0,5974578...	160,03767...	158,33091...	0,9581451...	1219,3293...	0,1971001...	0,9637120...	0,3762158...
9	191,91999...	159,49475...	0	78,819992...	0,9715883...	4,6865445...	0,5429229...	160,03767...	158,33091...	0,9650403...	1100,0110...	0,1765453...	0,9637120...	0,5552266...
10	191,92999...	159,46902...	0	78,829986...	0,9756999...	1,8515398...	0,5686492...	160,03767...	158,33091...	0,9654139...	1154,9976...	0,1857764...	0,9637120...	0,8109184...
11	191,80999...	159,30682...	0	78,709991...	0,9722147...	6,5992802...	0,7308502...	160,03767...	158,33091...	0,9388992...	1521,8281...	0,2605801...	0,9637120...	0,2176437...
12	191,69000...	159,32801...	0	78,589996...	0,9726182...	5,8502307...	0,7096557...	160,03767...	158,33091...	0,9432018...	1478,2490...	0,2519264...	0,9637120...	0,3054873...
13	192,08999...	159,66453...	0	78,989990...	0,9745424...	1,7748520...	0,3731384...	160,03767...	158,33091...	0,9785554...	766,19633...	0,1318832...	0,9637120...	0,9168507...
14	191,69999...	159,20605...	0	78,599990...	0,9704814...	6,0920275...	0,8316192...	160,03767...	158,33091...	0,9204438...	1719,1059...	0,3103964...	0,9637120...	0,1753104...
15	192,21000...	159,73333...	0	79,110000...	0,9690907...	5,8830688...	0,3043365...	160,03767...	158,33091...	0,9797853...	587,64259...	0,1149508...	0,9637120...	0,5528292...
16	192,36000...	159,86186...	0	79,259994...	0,9645009...	2,1316424...	0,1758117...	160,03767...	158,33091...	0,9802920...	388,01684...	0,1010010...	0,9637120...	0,9390238...

Figure 3.11: Output Unit of the GEDICorrect Pipeline

4

Methods

Having established the key components and design of GEDI Simulator and GEDICorrect, this chapter outlines the methodology, including the integration of these frameworks with the dataset and the specific experimental setup. Firstly, the Study Area is presented, along with the Data Collection conducted. After this, the usage of both frameworks is described. Finally, to evaluate the performance of the newly implemented framework, a set of experiments are defined to evaluate the GEDI geolocation correction method of GEDICorrect.

4.1 Geographic Study Area

The Study Area (named Abrantes) covers an area approximately 51.2 km long and 1.1 km wide in central Portugal located mainly in the district of Santarém (lat. 39.6 degrees N, long. 8.2 degrees W) and has approximately 4065.66 ha of forest, with a diverse land cover, characteristic of Mediterranean landscape. Dominant tree species include Hardwood forests (45.66%), Agriculture (19.61%), and Sparse Vegetation (10.90%).

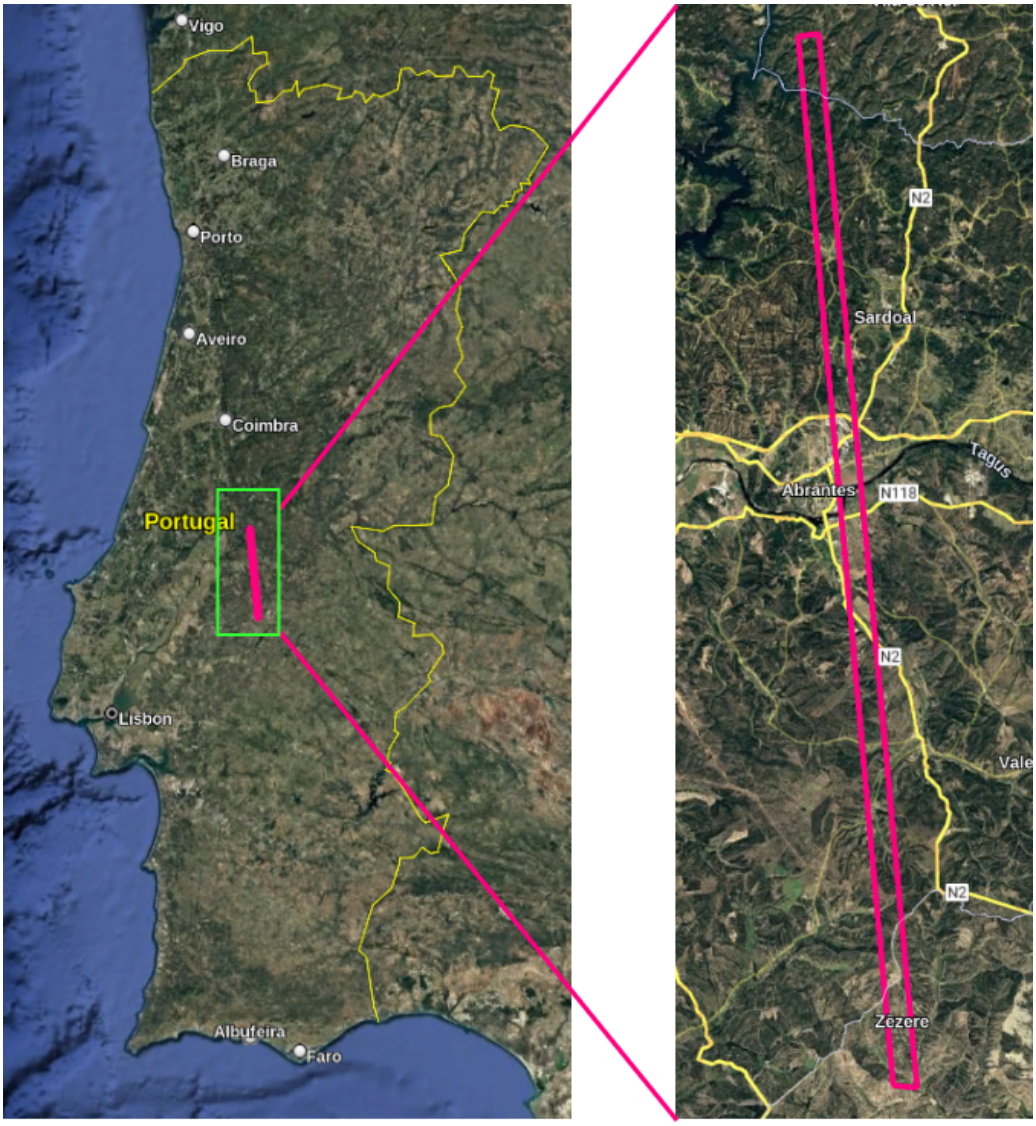


Figure 4.1: Study Area in Central Portugal. The bounds show the available ALS data.

4.2 Data Collection

4.2.1 ALS Data

The ALS data utilized in this project was acquired as part of the FUEL-SAT[1] project in November 2021, and has a nominal laser pulse density of 13.33 points/m^2. The data collection was conducted using a RIEGL VQ-1560i[2] sensor operating at 1064 nm (near-infrared) mounted on an Aero Commander 690A aircraft. The maximum pulse repetition rate was 2000 kHz, with a maximum scanning angle of $\pm$ 58.52°, and the average altitude during scanning was 936 meters above ground level. The company responsible for the ALS data acquisition, *TOPCAD Ingeniería S.L.*[3], provided the classified point cloud files in `.las` and `.laz` (compressed `.las`) formats. Figure 4.2 shows the point cloud distribution by elevation values after

[1] https://fuelsat.uevora.pt/
[2] http://www.riegl.com/nc/products/airborne-scanning/produktdetail/product/scanner/55/
[3] https://www.topcadingenieria.com/

automatic classification. Automatic classification refers to the process of categorizing or labeling the point cloud data captured by LiDAR sensors into distinct classes (e.g., ground, vegetation, buildings). Each point is classified with respect to its relative height, intensity, point density and return information (which help distinguish between vegetation and buildings based on the number of returns).

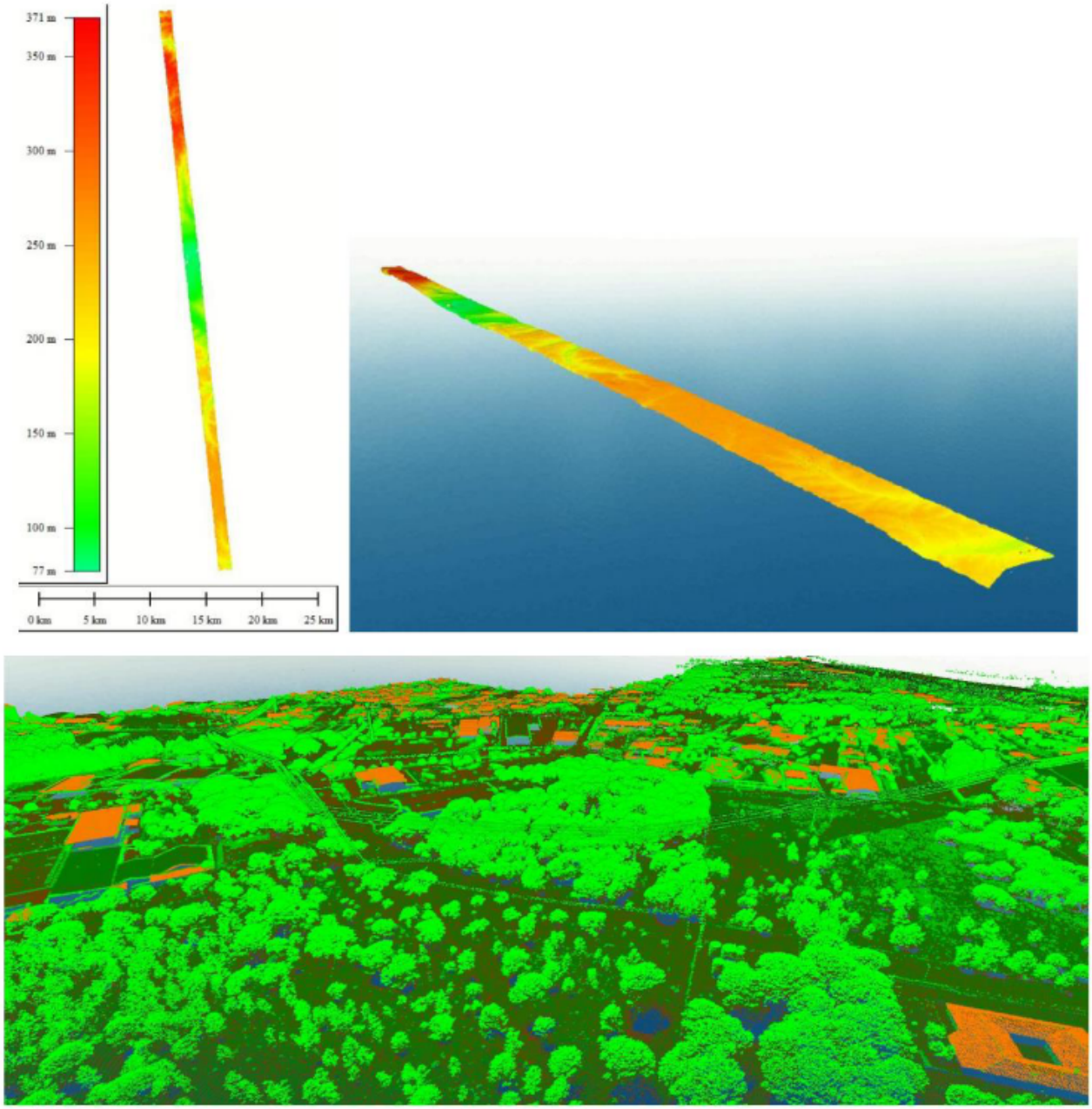


Figure 4.2: Visualization of the point cloud by elevation values (top) and after automatic classification (bottom).

In total, **118** `.las` **files**, with the Coordinate Reference System (CRS) set to the EPSG code of 3041, were used for geolocation correction methods.

4.2.2 GEDI Data

For the geolocation correction methods, both L1B and L2A (Version 2) data products from the GEDI mission were downloaded, clipped to fit the study area and extracted relevant variables. The L1B product provides geolocated corrected and smoothed waveforms and the L2A product provides waveform interpretation and derived products from each L1B received waveform, including ground elevation, canopy top height, and relative height (RH) metrics.

For this project, intersected GEDI orbits on our Study Area between **2019** and **2021** (from 01/01 to 31/12) were downloaded using the *GEDI-Pipeline* [Corado and Godinho, 2024]. This repository[4] provides an unified workflow for searching, downloading and processing GEDI data using NASA's Land Processes Distributed Active Archive Center (LP DAAC) through NASA's Common Metadata Repository (CMR), which simplifies the entire process into a single command.

The delivered GEDI data from the NASA's repository in HDF5 format was subsequently clipped to the Study Area for further analysis. The output of *GEDI-Pipeline* included 47 intersected GEDI orbits in *Geopackage* (.GPKG) files for each selected data product (L1B and L2A), resulting in a total of 97 files. The following command was used to execute the *GEDI-Pipeline*:

```
$ python3 gedi_pipeline.py --dir ./FUELSAT --product GEDI01_B --version 002 --start
↪  2019.01.01 --end 2021.12.31 --roi 39.661139,-8.212886,39.199983,-8.139717
```

This same command was repeated for the "`GEDI02_A`" data product by adjusting the `--product` parameter.

Although the ALS data was collected in 2021, the inclusion of GEDI data from previous years not only increased the dataset size to benchmark the different frameworks, but also allowed for a more comprehensive study of forest structure through time and ensured that anomalies in the dataset (e.g. changes caused by natural or anthropogenic factors) were accounted for further analysis and filtering.

Considering the noise and uncertainty in GEDI data, a preprocessing step is required to ensure that only high-quality footprints will be used for the geolocation correction.

Preprocessing of the GEDI Data

The framework implemented in this dissertation requires a merged dataset comprising both the L1B and L2A data products. Since each footprint is unique and contains an identifier (*shot_number*), the merging process involved aligning the filtered outputs from *GEDI-Pipeline* based on the *shot_number* variable and selecting the complete RH profile from the L2A product. This process resulted in 47 merged files, which will be referred to as *GEDI_CorrectTest* for the remainder of the study. Additionally, the full code for the applied data processing and merging both L1B and L2A is provided in Appendix A.2.

Each footprint's waveform (*rxwaveform* variable) consists in a list of values at each height bin (Z), which Z is calculated by subtracting the highest elevation return (Z0) to the ground return (ZG). The size of this list is the number of elevation bins (described by the `rx_sample_count`).

To ensure the use of only high-quality GEDI footprints, a set of quality metrics developed and suggested by the GEDI Science Team and community were used:

[4] https://github.com/leonelluiscorado/GEDI-Pipeline

- ***degrade_flag*** $== 0$ - indicates a low probability of degraded geolocation under suboptimal operating conditions [Roy et al., 2021];

- ***quality_flag*** $== 1$ - indicates that the given footprint meets quality criteria in terms of energy, sensitivity, amplitude, and real-time surface tracking [Hofton et al., 2019];

- ***solar_elevation*** < 0 - This metric is utilized to determine whether GEDI footprint acquisitions occur during night or day. Only the nighttime acquisitions were retained for analysis, as indicated by a solar elevation angle less than 0 [Beck et al., 2021];

- ***sensitivity*** > 0.9 - Sensitivity refers to the maximum canopy cover that the GEDI laser shots can penetrate, considering the Signal to Noise Ratio (SNR) of the waveform. Based on previous studies that assess the impact of sensitivity on GEDI footprint accuracy [V.C. Oliveira et al., 2023], in this work, only footprints with a sensitivity greater than 0.90 were selected;

- ***(RH95*** $>= 5$ ***&& num_detected_modes*** $== 1$***)*** - This custom filter ensures that in all footprints representing forests (RH95 higher than 5 meters), the waveform generated by GEDI measurements exhibits more than one mode. Typically, a tree's waveform contains at least two modes: one corresponding to the canopy and another to the ground;

- ***RH95*** $<= 30$ - This custom criterion aims to eliminate erroneous canopy height measurements resulting from various factors (such as electric lines, aerosols, etc.) that interact with the GEDI LiDAR signal. For this case in Portugal, trees above 30 meters are rare, if they exist at all;

- $|$ ***elev_lowestmode*** - ***digital_elevation_model*** $| <= 50m$ - To eliminate footprints with erroneous ground detection, all footprints with an absolute difference between the elevation of the lowest mode (elev_lowestmode) and the TanDEM-X elevation at the GEDI footprint location (digital_elevation_model) greater than 50 meters were excluded from the analysis.

After applying these filters to the *GEDI_CorrectTest* dataset, a total of **1956** footprints were retained for the subsequent analysis. This represented a reduction of approximately 63.11% compared to the original dataset. Some orbits (files) were entirely excluded for not meeting the required conditions, leaving a total of 18 files in the final dataset. Figure 4.3 provides a visual comparison between the original data with the *GEDI_CorrectTest* dataset. Finally, for the entire *GEDI_CorrectTest* dataset, the geolocation coordinates of each footprint were adjusted to match the CRS of **EPSG:3041**, ensuring that both CRS from GEDI and the ALS's data are aligned.

The final structure of the *GEDI_CorrectTest* dataset consists of 18 files, containing a total of 1956 high-quality footprints, ready for further geolocation correction and analysis in GIS tools. Additionally, the dataset was converted into HDF5 format, mimicking the hierarchy from the originally downloaded GEDI data (as provided by NASA LP DAAC), which is designed to serve as the input for geolocation correction frameworks such as the GEDI Simulator's `collocateWaves`. The HDF5 version of the dataset merged the 18 `.GPKG` file into 1 `.H5` file.

This dual-format approach, with both GPKG and HDF5 versions, allows for flexibility in subsequent analysis and ensures that the dataset is suitable for integration into various workflows.

Figure 4.4 depicts the variables extracted for one footprint from the *GEDI_CorrectTest* dataset, including both L1B (e.g., *rxwaveform*) and L2A variables (e.g., *rh* metrics).

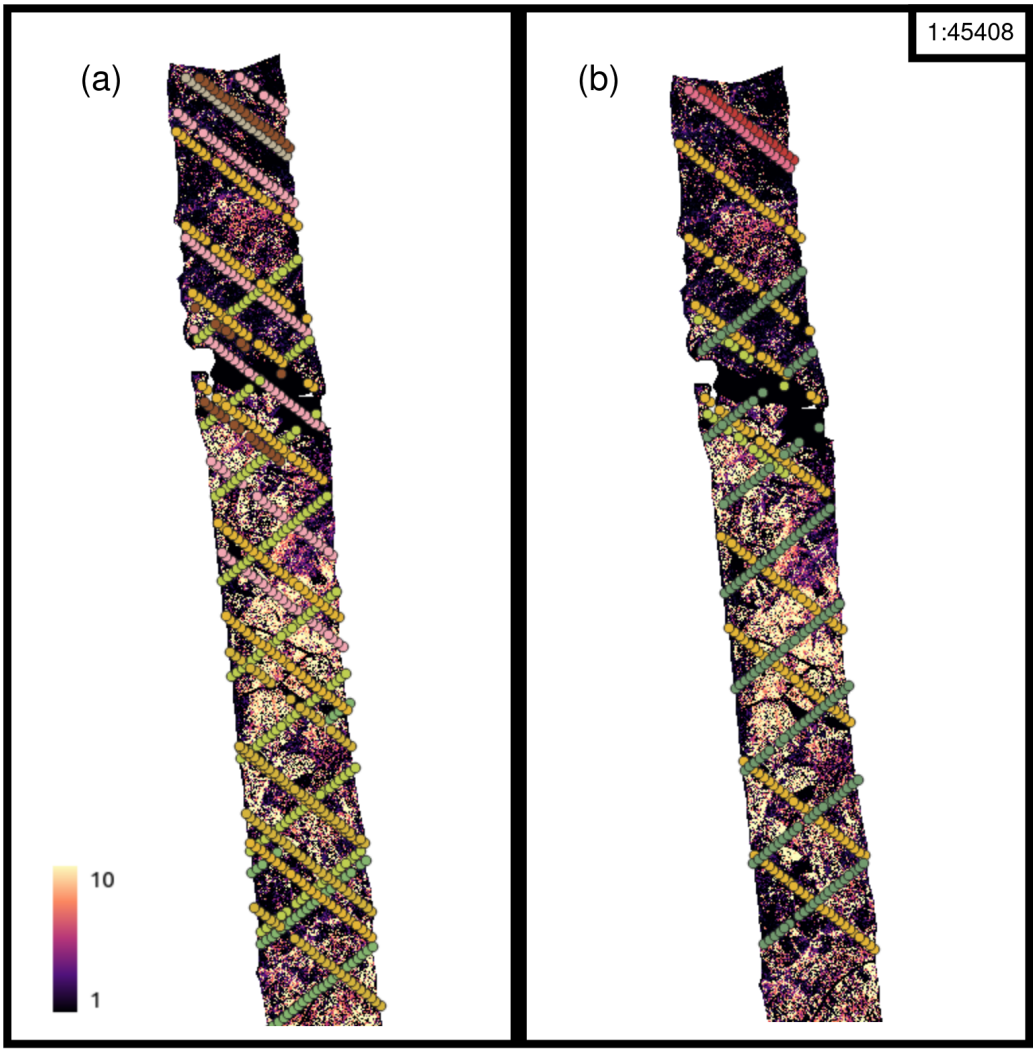


Figure 4.3: Visual representation of the footprints over the top part of the Study Area from the original downloaded data (a) and the filtered *GEDI_CorrectTest* dataset (b).

4.3 Infrastructure

Before describing the execution details of both frameworks, a brief detail of the infrastructure is needed. The programs and experiments were tested using the NIIAA Cluster. The machine is equipped with two Intel(R) Xeon(R) Silver 4110 processors, featuring a total of 32 CPU threads distributed across 16 cores (8 cores per processor with 2 threads per core), operating at a base clock speed of 2.10 GHz and a maximum of 3.00 GHz. The system's memory capacity totals 88 GB and the operating system used was Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-116-generic x86_64). Additionally, `tmux` was employed to manage multiple jobs across sessions.

4.4 Usage of GEDI Simulator

This section focuses on the compilation and execution instructions for GEDI Simulator, that were used in initial tests and the main experiments described in Section 5.

Feature	Value
BEAM	BEAM0101
shot_number_x	51860500200152196
Latitude	39,39983160094569
Longitude	-8,171757913827445
index	38108
geolocation_degrade	0
geolocation_delta_time	58748692,40574765
geolocation_digital_elevation_model	120,975341796875
geolocation_elevation_bin0	168,16713280789554
geolocation_elevation_lastbin	57,2971149655059
geolocation_local_beam_elevation	1,5398293733596802
geolocation_shot_number	51860500200152196
geolocation_solar_elevation	-63,070396423339844
land	1
ocean	0
sea_ice	0
land_ice	0
inland_water	0
noise_mean_corrected	205,875
rx_sample_count	741
rx_sample_start_index	54113361
rxwaveform	207.09981,207.9413,208.96637,209...
shot_number_y	51860500200152196
stale_return_flag	0
tx_sample_count	128
txwaveform	205.41916,205.4693,205.54492,205...
date	2019/11/11
shot_number	51860500200152196
degrade_flag	0
quality_flag	1
elev_lowestmode	121,08483123779297
digital_elevation_model	120,975341796875
num_detectedmodes	1
solar_elevation	-63,070396423339844
sensitivity	0,9012316465377808
rh_1	-3,3299999237060547
rh_2	-3,140000104904175
rh_3	-2,990000009536743
rh_4	-2,8399999141693115
rh_5	-2,7300000190734863
rh_6	-2,619999885559082

Figure 4.4: Example footprint (shot_number: 51860500200152196) from the *GEDI_CorrectTest* dataset. The L1B variables (e.g. *rxwaveform*) are merged with L2A variables (e.g. *rh*)

4.4.1 Compiling GEDI Simulator

There are three ways to compile the project:

- **Singularity Container** - Uses a script provided by the repository that sets up a container with all the programs installed;
- **Compile from Source** - Clones the repository and installs the required packages manually;
- **Bash Compilation Script** - Automatically runs the step *Compile from Source* and creates the necessary directories.

Singularity[5] is a containerization platform that allows the user to create portable and reproducible environments. Each environment acts as an isolated operating system that can be shared across a multitude of systems. The Singularity Setup script sets up this container by downloading and configuring all the required packages encapsulated in a Docker image based on Fedora 28. GEDI Simulator's repository contains a

[5] https://docs.sylabs.io/guides/3.5/user-guide/introduction.html

makeSingularity.txt[6] file which contains a set of instructions to create the container. Although this method is fail-proof, the created environment is outdated and may contain security vulnerabilities, which could compromise the performance and the accuracy of the experiments.

In the case of compiling from source, the process requires to manually clone the repository and install each required package on the system. The required packages are:

- `GNU Scientific Library` [Gough, 2009]
- `Geotiff` [Niles Ritter, 2000]
- `HDF5` [HDFGroup, 1987]
- `GDAL` [Rouault et al., 2024]
- `CMPFIT` [Craig B. Markwardt, 2022]

Additionally, it requires the installation of `C-tools`[7] and `libClidar`[8] from Steven Hancock's repository. Depending on the user's operating system, it may require installation from specific package managers and compilation steps for compatibility purposes. Once the required packages are installed, the final step consists of making use of the available `Makefile` to compile and install each program in the project. While this method offers flexibility, it requires more manual effort to configure the environment correctly. Fortunately, a script that automates such process is available, significantly reducing setup time.

For this study, the bash compilation script was used. The following code demonstrates on how to fetch the compilation script, give it executable permissions and running the script:

```
$ wget https://bitbucket.org/StevenHancock/gedisimulator/src/master/installGedi.bash
$ chmod +x installGedi.bash
$ ./installGedi.bash
```

During the initial execution, the installation script failed to set up properly due to an incorrect path specified for the `HDF5_LIB` variable, which is from a required package HDF5. Updating this variable to point to the host's HDF5 installation resolved the issue. Appendix A.1 provides the updated bash script used for the experiments. After this adjustment, the installation completed successfully, and GEDI Simulator was ready for use.

4.4.2 Executing GEDI Simulator

The GEDI Simulator repository provides example commands and descriptions of each program's options, previously detailed in Section 3.1.

`gediRat`

The `gediRat` program simulates GEDI-like waveforms at specific coordinate locations using the input ALS point-cloud data. The following command demonstrates how to simulate a GEDI-like waveform at a single coordinate (in the same CRS as the ALS data), with the waveform cropped at 500 height bins:

[6] https://bitbucket.org/StevenHancock/gedisimulator/src/master/makeSingularity.txt
[7] https://bitbucket.org/StevenHancock/tools/src/master/
[8] https://bitbucket.org/StevenHancock/libclidar/src/master/

```
$ gediRat -inList alsList.txt -coord 595208.08 4269772.19 -maxBins 500 -output
↪  waveform.txt
```

The next command demonstrates how to simulate multiple waveforms from a list of coordinates, outputting the results in HDF5 format:

```
$ gediRat -inList alsList.txt -listCoord coords.txt -hdf -output waveform.h5
```

The list of coordinates is provided in an ASCII file, with each set of coordinates separated by a newline. An example structure for the coordinates file is shown below:

```
595208.08 4269772.19
595208.08 4269784.22
595210.78 4270001.75
...
```

Both output files (`waveform.txt` and `waveform.h5`) can be processed with the `gediMetrics` program to extract relevant waveform metrics such as ground slope, elevation, canopy cover, and, most importantly, the RH profile. The RH profile is used in the CRSSDA method described in *GEDICorrect* Unit 3 (see Section 3.2.1).

`gediMetrics`

The `gediMetrics` program generates waveform metrics from real or simulated GEDI L1B data (produced by `gediRat`), offering a high level of customization. For the initial experiments, default settings were used for parameters such as `-varScale`, `-sWidth`, `-rhRes`, and `-laiRes` (as described earlier in Section 3.1.2).

The following command demonstrates the command used to extract waveform metric information from a simulated waveform file generated by `gediRat`:

```
$ gediMetrics -input waveform.h5 -readHDFgedi -ground -varScale 3.5 -sWidth 0.8 -rhRes 1
```

The output of `gediMetrics` can be further processed into structured formats, such as a *pandas* DataFrame or a CSV file. However, GEDI Simulator does not provide tools for parsing these results, leaving it up to the user to organize the output as needed. Additionally, `gediMetrics` can be used after collocating footprints with the `collocateWaves` program, allowing for the simulation of corrected footprint data.

`collocateWaves`

The `collocateWaves` program is GEDI Simulator's approach to footprint geolocation correction. Its objective is to find the optimal (X, Y, Z) placement vector that maximizes the correlation between reported

and simulated GEDI waveforms for an entire orbit. To use this program, the user requires two inputs: GEDI data files in HDF5 format and a list of ALS files covering the target area, provided in an ASCII file with the absolute paths of the `.las` files. The following command demonstrates the most efficient way to identify the ground offset between GEDI and ALS, aligning the GEDI files with ALS data using the **Bullseye** mode:

```
$ collocateWaves -listALS alsList.txt -listGedi gediList.txt -readHDFgedi -aEPSG 4328
↪  -solveCofG -geoError 30 5 -writeWaves simulated.h5 -minDense 3 -minSense 0.9
```

For the initial Baseline Assessment experiments (described in Section 5.1), the HDF5 version of the *GEDI_CorrectTest* dataset was used with the following command:

```
$ collocateWaves -listALS FUELSAT_AREA.txt -listGedi GEDI_FUELSAT_FILES.txt -readHDFgedi
↪  -aEPSG 32629 -simplex -solveCofG -geoError 30 5 -writeWaves COLLOCATED_FUELSAT.h5
↪  -minDense 3 -minSense 0.9
```

This command executes `collocateWaves` in *Simplex* mode, as detailed in Section 3.1. The output (`COLLOCATED_FUELSAT.h5`) was subsequently processed using `gediMetrics` with the following command:

```
$ gediMetrics -input COLLOCATED_FUELSAT.h5 -readHDFgedi -ground -varScale 3.5 -sWidth
↪  0.8 -rhRes 1 -outRoot "FUELSAT_"
```

4.5 Implementation of GEDICorrect

GEDICorrect was implemented in Python 3.12 within an Anaconda-managed virtual environment to ensure consistent dependencies across different systems. Key libraries used include *numpy* and `pandas` for data manipulation, *geopandas* for handling geospatial data, *shapely* for geometric operations such as creating buffers and *laspy* to read the ALS data.

File handling and external program execution are managed using the *os* and *subprocess* libraries, while the *multiprocessing* library and its `Pool` class were employed to enable parallel execution of the footprint simulations, optimizing performance for GEDICorrect. Additionally, careful memory management practices were implemented, such as using `del` to remove unnecessary dataframes and output lists between input GEDI files, ensuring efficient resource utilization during the entire execution. Finally, for plotting results in the following sections, the *matplotlib* and *seaborn* libraries were used.

4.5.1 Parallel Techniques

To enhance efficiency and reduce the overall runtime, the framework incorporates parallel processing techniques, ensuring that multiple footprints can be processed at the same time.

Since this geolocation correction method operates at the footprint level, each footprint undergoes through Unit 2 (Simulation) and Unit 3 (Scoring) described in Section 3.2.1. For this, a multiprocessing pool

was implemented for parallel processing. This pool spawns N processes, assigning to each one an equally divided block of input footprints to process. To do this, the `pool.imap_unordered()` function is used.

The user can also select the desired number of processes. If no number is selected but parallelization processing is, the framework selects the maximum number of processes minus 2 to keep the system responsive (using `OS.CPU_COUNT()`-2).

In the Simulation Unit, for I/O handling across multiple calls to the GEDI Simulator programs (`gediRat` and `gediMetrics`), a `TemporaryDirectory` is created, where each process handles its own processed files. This approach prevents program output conflicts, ensuring that data generated by one process does not interfere with another. For example, the Latitude and Longitude coordinates needed by `gediRat` are saved in a unique temporary location for each process. Moreover, each output file is prefixed with the process ID, which further ensures the data being isolated from other process. After processing, the temporary directory is deleted. An example of the temporary directory structure is shown in Listing 4.1.

```
temp_dir/
        alsList.txt
        3001/
            3001.metric.txt
            3001_simulated.h5
        3002/
            ...
        3003/
            ...
```

Listing 4.1: Example of TemporaryDirectory structure

Finally, each process begins with a unique random seed for generating points around the footprint, ensuring that the same random seed from the master process is not passed down to child processes[9]). This approach guarantees that no two or more processes produce the same distribution of candidate geolocation points.

Once the simulation step is completed, the *Pool* merges the simulation results into a list, so that the results from each process are introduced in the Scorer class for further processing. The same mechanism used in *Pool* is also used in the Scoring step, where each process is assigned a set of simulated footprints to score.

By implementing these simple parallel techniques, GEDICorrect can achieve a significant reduction in processing time, allowing for the geolocation correction of large GEDI datasets within a more reasonable timeframe, without compromising the accuracy of the results.

4.6 Usage of GEDICorrect

The execution of the newly implemented framework is managed through a single Python script (`gedi_correct.py`), which creates a `GEDICorrect` object and applies the selected geolocation correction method based on user-defined settings. These user-defined settings, also referred to as program arguments, allow for customization of the correction process.

Some of the most important command options for `gedi_correct.py` include:

- `--granules_dir` - Specifies merged L1B-L2A GEDI input file directory for batch correction;

[9] https://github.com/numpy/numpy/issues/9650

- `--input_file` - Specifies a single merged L1B-L2A GEDI input file for correction;
- `--las_dir` - Specifies the directory of `.las` files required for processing and simulation which must overlap with the input granule file(s);
- `--out_dir` - Specifies the directory in which to save the corrected input granules and simulated points;
- `--save_sim_points` - Flag option to save all the simulated points around each footprint;
- `--save_origin_location` - Flag option to save the original location simulated footprint;
- `--criteria` - Enumerates the set of criteria for best simulated footprint selection, based on the list [`"wave"`, `"rh_distance"`, `"kl"`, `"terrain"`, `"wave_distance"`], or the "`all`" to select all criteria;
- `--n_points` - Specifies the number of points to simulate around each input footprint, which defaults to 100;
- `--radius` - Specifies the maximum distance for radius to simulate points around each original footprint, defaulting to 12.5 meters;
- `--min_dist` - Specifies the minimum distance between simulated points around each original footprint, defaulting to 1 meter;
- `--parallel` - Flag option to run *GEDICorrect* in parallel with "`--n_processes`" processes. If no number is defined, it defaults to all available system's processes minus 2;
- `--n_processes` - Specifies the number of processes to use for parallel processing, if the "`--parallel`" option is activated. This is optional but it allows users to control the number of processes used.

In summary, the default settings for the standard geolocation correction method in *GEDICorrect* are as follows: i) 100 points simulated around each reported footprint; ii) points simulated up to 12.5 meters from the reported footprint; iii) a minimum spacing of 1 meter between simulated points.

The following command demonstrates how to execute the `gedi_correct.py` script in parallel with 16 processes, using all available criteria:

```
$ python3 gedi_correct.py --granules_dir /FUELSAT_FULL_TEST_FILTERED/ --las_dir
↪  /data/lcorado/ALS/las_com_CRS/ --out_dir /data/lcorado/FUELSAT_CORRECTED
↪  --save_sim_points --criteria "all" --parallel --n_processes 16
```

For this work, this command was extensively modified to suit each experiment's specific requirements. Additionally, the full code for the `gedi_correct.py` script is provided in Appendix A.4.

4.7 Accuracy Assessment Metrics

In order to evaluate GEDICorrect's performance, a range of metrics will be used to assess both the geolocation correction accuracy and computational efficiency. The following metrics will guide the evaluation steps described in the next section (Section 4.8):

- R-squared coefficient (R^2)
- Root Mean Squared Error (RMSE)
- Mean Average Error (MAE)
- Speedup (S)
- Amdahl's Law

Given the simulated RH95 values of the corrected footprints ($RH95_{sim}$) and the reported RH95 values for each respective footprint ($RH95_{orb}$) the following metrics were used.

R^2, depicted in Equation 4.1 measures the the proportion of variance in the dependent variable that can be explained by the independent variable, varying (usually) from 0 to 1, with 0 implying no correlation; RMSE, given by Equation 4.2, calculates the average magnitude of the error between two variables, penalizing larger errors more severely, which could help identify significant deviations in geolocation; MAE, calculated through Equation 4.3, represents the average of the absolute errors between reported and simulated RH95 values, which is a more interpretable measure of the overall geolocation error. In the Equations $RH95\_sim$ is the simulated RH95 of the vector of the corrected points, $RH95\_orb$ is the reported RH95 vector, m_{RH95_orb} is the mean of the reported RH95 vector and n is the size of both RH95 vectors.

$$R^2 = 1 - \frac{\sum(RH95\_orb_i - RH95\_sim_i)}{\sum(RH95\_orb_i - m_{RH95\_orb})} \tag{4.1}$$

$$RMSE = \sqrt{\frac{\sum(RH95_{orb} - RH95_{sim})^2}{n}} \tag{4.2}$$

$$MAE = \frac{\sum |RH95_{orb} - RH95_{sim}|}{n} \tag{4.3}$$

To assess the improvement in runtime of *GEDICorrect* when it is executed in parallel with N processes, the *Speedup*, given by Equation 4.4, can be calculated with reference to the program's elapsed time (T). While $S = 1$ shows no speedup, $S > 1$ indicates an improvement in performance when parallelizing the program.

$$S = \frac{T_{sequential}}{T_{parallel}} \tag{4.4}$$

Finally, a theoretical speedup can be predicted using Amdahl's Law [Amdahl, 1967], which calculates the maximum improvement in performance given a portion of the program that can be parallelized, and is expressed as Equation 4.5, where P is the proportion of the program that can be parallelized (between 0 and 1) and N is the number of processes or used for parallel execution.

$$S_{max} = \frac{1}{(1 - P) + \frac{P}{N}} \tag{4.5}$$

According to Amdahl's Law, even with an infinite number of processes, the speedup is ultimately limited by the fraction of the program that cannot be parallelized ($1 - P$). Specifically, it assumes that a portion of the program must still be executed serially, which limits the overall speedup. Therefore, this metric helps determine how much parallelization will benefit GEDICorrect, and determine the extent to which each step can be parallelized.

4.8 Experiments

To test the newly designed system, a set of four experiments was defined to evaluate the accuracy and performance of *GEDICorrect* and directly compare it to GEDI Simulator. The GEDI RH95 metric, which has been utilized in several recent GEDI studies [Potapov et al., 2021, Roy et al., 2021, Lahssini et al., 2022], was adopted as the representative measure of canopy height in this work.

The four experiments are as follows:

Baseline Assessment - The Baseline Assessment experiment will:

1. Evaluate the accuracy of on-orbit RH95 ($RH95_{orb}$) by comparing it with the simulated RH95 ($RH95_{sim}$) that was derived from the ALS data at the original GEDI location (i.e. without any geolocation correction method);
2. Assess the improvement accuracy in $RH95_{orb}$ after applying the `collocateWaves` geolocation correction method;
3. Assess the improvement accuracy in $RH95_{orb}$ after applying the Pearson's and Spearman correlation criteria within *GEDICorrect*.

Criteria Assessment - The Criteria Assessment aims to test the feasibility of the newly proposed metrics within the Scorer Unit (Unit 3). Specifically, this experiment will:

1. Evaluate the accuracy of each individual criterion by evaluating the highest R^2 between $RH95_{orb}$ and $RH95_{sim}$;
2. Perform a grid search across various combinations of criteria to identify the optimal configuration that achieves the highest accuracy (i.e., the highest R^2 between $RH95_{orb}$ and $RH95_{sim}$).

Efficiency Assessment - This experiment aims to assess the overall efficiency of *GEDICorrect*, focused on both optimized coding practices and the use of parallelization for large-scale datasets. To carry out this assessment, the experiment will:

1. Evaluate the runtime of both frameworks (GEDI Simulator and *GEDICorrect*) when run sequentially;
2. Quantify the theoretical speedup of *GEDICorrect* using Amdahl's Law;
3. Measure the efficiency of the parallelization methods employed within *GEDICorrect* by performing a grid search across a number of processors.

Points Distribution Assessment - Finally, the quantity and randomness of the simulated points around each footprint are tested. This assessment is divided into two tests:

1. *Simulated Points Assessment* - Determine the optimal number of randomly generated points around each footprint, considering the trade-off between accuracy and computational costs;

2. *Stochasticity Assessment* - Assess the impact of the randomness in point generation on determining the best geolocation for each footprint.

Each experiment builds upon the previous one, creating a progressive structure. Through these experiments, the ultimate goal is to identify the optimal configuration and the most efficient method for geolocation correction with *GEDICorrect*.

5

Results & Discussion

This chapter presents the main results obtained for each proposed experiment (see section 4.8). The dataset used for the experiments, as previously described in Section 4.2, includes the ALS point cloud data of the presented study area along with its corresponding cleaned and filtered GEDI dataset, the *GEDI_CorrectTest* dataset. For all experiments, except "*Simulated Points Assessment*", the number of generated points around each footprint was 100, which is *GEDICorrect*'s default option (see Section 4.6).

5.1 Baseline Assessment

The baseline test for this experiment using GEDI Simulator's `collocateWaves` was performed with the HDF5 version of *GEDI_CorrectTest* described in Section 4.2. The *gediMetrics* program was used to extract the RH profile from the program's output, and the results were formatted into a DataFrame for further analysis. The goal was to assess the effectiveness of this standard geolocation correction process in improving the accuracy of the GEDI footprints.

The results of this initial test show a moderate relationship ($R^2 = 0.52$) between the reported ($RH95_{orb}$) and simulated ($RH95_{sim}$) GEDI canopy height at the original GEDI location, i.e., without any geolocation

correction. This result highlights the impact of the geolocation error in GEDI data, which is estimated to be around 10 meters [Dubayah et al., 2020]. This moderate correlation is similar to what have been reported in other relevant studies focused on the assessment of the impact of geolocation error on GEDI canopy height accuracy [Roy et al., 2021]. In areas where the landscape is dominated by heterogeneous land cover types, such as the study area addressed in this dissertation, the impact of the geolocation error on GEDI measurements is more evident. Figure 5.1 shows the relationship between reported ($RH95_{orb}$) and simulated ($RH95_{sim}$) GEDI canopy height at the reported location.

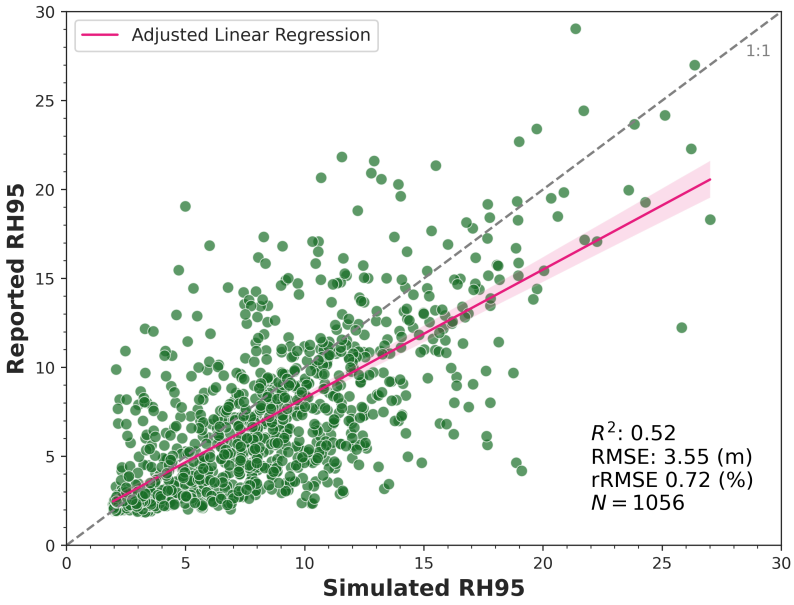


Figure 5.1: Relationship between reported ($RH95_{orb}$) and simulated ($RH95_{sim}$) GEDI canopy height at the original GEDI reported location.

The `collocateWaves` method, which represents the standard GEDI Simulator geolocation correction process, showed no significant improvement in GEDI canopy height accuracy ($R^2 = 0.52$), indicating that the orbit-level process in this study area is not a viable option for improving geolocation accuracy. This result was also recently reported by East et al. [East et al., 2024], who found no significant improvement in the accuracy of GEDI RH98 (used as canopy height metric) after geolocation correction using the `collocateWaves` tool. They reported an RMSE of 5.32 without geolocation correction and 5.64 meters with geolocation correction. Figure 5.2 shows the relationship between reported ($RH95_{orb}$) and simulated ($RH95_{sim}$) GEDI canopy height after geolocation correction using `collocateWaves`.

One of the goals within this first experiment, was to compare the accuracy of *GEDICorrect*'s geolocation correction at the footprint level against the orbit-level correction of `collocateWaves`. For this, the '*wave*' Scoring criterion was used, testing two different correlation methods (Pearson and Spearman). Surprisingly, and as can be seen in Table 5.1, the footprint-level correction approach implemented in *GEDICorrect* achieved a similar accuracy as observed in the GEDI Simulator geolocation correction method. The footprint-level approach resulted in an R^2 of 0.5232 using the Pearson correlation metric, while the baseline orbit-level GEDI Simulator produced an R^2 of 0.51. This result may be explained by the fact that when using `collocateWaves` in a small study area, i.e., with only a few thousand GEDI footprints representing limited terrain and vegetation conditions, the geolocation correction accuracy tends to be similar to that of the footprint-level approach. It's important to note that `collocateWaves` computes a constant (X, Y, Z) offset that will be used to shift all the footprints according to this offset. Therefore, the larger the orbit (i.e., the number of footprints), the less accurate this constant becomes for correcting footprints that are

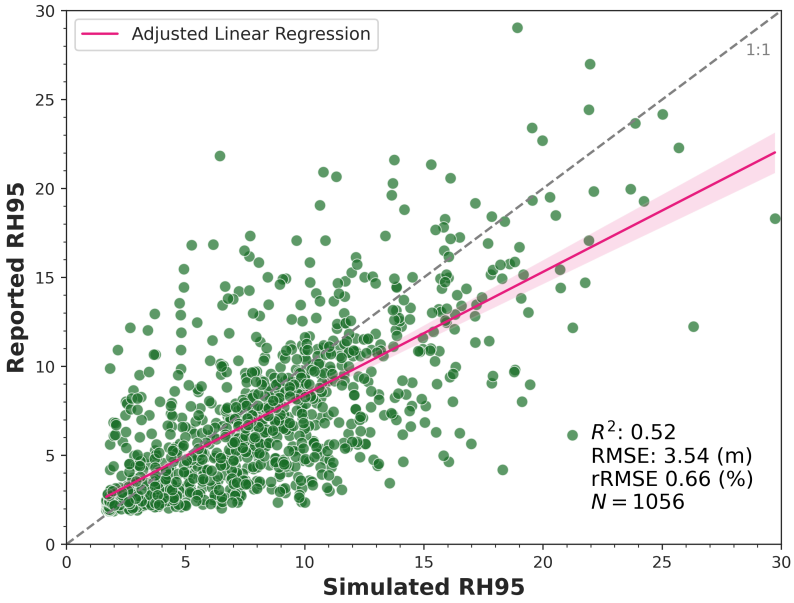


Figure 5.2: Relationship between reported ($RH95_{orb}$) and simulated ($RH95_{sim}$) GEDI canopy height after geolocation correction using `collocateWaves`

representing a more diverse landscape. The experiment using Spearman's correlation metric yielded a very similar R^2 value compared to Pearson's correlation (0.5161 vs. 0.5232), indicating that both metrics can be applied for this task. However, while Pearson's correlation is more suitable for linear relationships, this is not typically the case for LiDAR waveforms. Based on available studies, there is no research specifically focused on using Spearman's correlation to assess LiDAR waveform similarity, although its use in other waveform similarity studies (e.g. electrical current waveforms) has been reported [Rebonatto et al., 2017]. In summary, *GEDICorrect* defaults to using Spearman's correlation for the '*wave*' criterion, as it is more appropriate for non-linear data, such as waveforms.

Method	Correlation	Criteria	R^2
GEDI Simulator	Pearson	-	0.52
GEDICorrect	Pearson	*wave*	0.5232
GEDICorrect	Spearman	*wave*	**0.5161**

Table 5.1: Accuracy Assessment using different correlation based methods, Pearson and Spearman.

5.2 Criteria Assessment

GEDICorrect improves upon GEDI Simulator by adding additional criteria to compare simulated and reported waveforms. In this work, four new criteria were introduced to improve the accuracy of footprint geolocation. Table 5.2 presents the accuracy results of the GEDI canopy height (RH95) after applying the implemented geolocation correction criterion.

It can be seen that the `rh_distance` criterion has the greatest impact on geolocation accuracy ($R^2 = 0.86$), followed by the `kl` criterion ($R^2 = 0.62$), which also demonstrates the superiority of the KL metric as a method for waveform curve similarity when comparing different waveforms (i.e., reported vs. simulated) [Zhou et al., 2016]. When comparing the waveform correlation-based matching (`wave`) approach with the distance-based approach (`wave_distance`), it is evident that the distance-based method produces

Criterion	R^2	RMSE (m)	MAE (m)
wave	0.5161	3.34	2.25
wave_distance	0.6020	3.29	2.40
kl	0.6232	3.19	2.31
terrain	0.5121	3.54	2.52
rh_distance	**0.8604**	1.68	1.09

Table 5.2: Single criterion evaluation on GEDICorrect

better results ($R^2 = 0.60$ vs $R^2 = 0.52$). This can be explained by the fact that the correlation method captures the overall similarity between the two waveform curves rather than their alignment. The curves may have a similar shape and produce a high correlation score, yet still be far from each other, meaning they are not perfectly overlapping or aligned. In contrast, a distance-based criterion, which computes the cumulative absolute difference between the reported and simulated waveform energy values, may be a more direct approach to quantify the degree of waveform curve overlap. In summary, correlation-based methods tend to capture similarity in the shape of two curves, while distance-based methods focus on their alignment and magnitude differences (e.g. [Shirkhorshidi et al., 2015]). Finally, regarding the `terrain` criterion, the results show no improvement in GEDI canopy height accuracy after applying the terrain matching geolocation correction method. This outcome may reflect the impact of the initial data filtering process, where all GEDI footprints with absolute differences greater than 10 meters between the GEDI-reported terrain elevation (`elev_lowest_mode`) and the elevation from the SRTM Digital Elevation Model were removed.

GEDICorrect also supports combining multiple criteria for a more thorough evaluation, as described in Section 3.2.1. When using multiple criteria, the final score of each simulated footprint is calculated as the equally weighted mean of each criterion's score. To identify the optimal combination of criteria, a grid search was performed across all of the possible combinations. Table 5.3 displays the results of these combinations.

The results show that the '`rh_distance`' metric positively impacts the performance of the geolocation correction, consistently producing the highest R^2 values across both individual and combined tests. This suggests that the RH profile is a reliable indicator of footprint geolocation accuracy, as the variability in RH values across the profile captures key aspects of the GEDI full waveform energy (e.g., [de Conto et al., 2024]). Overall, these results confirm the usefulness of comparing reported and simulated RH profiles as a metric of GEDI geolocation error, as observed in Jia et al. [Jia et al., 2024] who used the absolute difference between reported and simulated RH profile to measure the impact of geolocation error on the reliability of the GEDI Biomass product. However, the use of RH profile differences, such as the `rh_distance` metric, as a criterion for GEDI geolocation correction has never been tested, highlighting the pioneering nature of this work.

Although the highest performing observed combination is the individual `rh_distance`, the `kl+rh_distance` combination was selected as the criterion for GEDI geolocation correction in this work due to their complementary nature - `kl` focuses on waveform curves, and `rh_distance` targets the RH profile. Figure 5.3 shows the relationship between reported ($RH95_{orb}$) and simulated ($RH95_{sim}$) GEDI canopy height after geolocation correction using the `kl+rh_distance` criterion.

Additionally, Figure 5.4 illustrates the varying results produced by different methods for GEDI geolocation correction. `collocateWaves`'s method slightly adjusts the original footprint location by approximately 1.5 meters. In contrast, the method proposed here, which combines KL-based waveform matching with the CRSSDA on the RH Profile approach, shifts the original GEDI footprint by a more substantial distance of 7.6 meters, resulting in a high similarity between the original and simulated waveforms (see Figure 5.5).

Criteria Combination	R^2	RMSE (m)	MAE (m)
rh_distance	0.86	1.68	1.09
rh_distance + wave_kl	0.82	1.91	1.20
rh_distance + wave	0.82	1.89	1.22
rh_distance + wave_kl + wave	0.81	2.00	1.26
rh_distance + wave_distance	0.80	2.02	1.30
rh_distance + wave_kl + terrain	0.80	2.07	1.32
rh_distance + wave_distance + wave_kl	0.79	2.11	1.35
rh_distance + wave_distance + wave_kl + wave	0.79	2.12	1.35
rh_distance + wave_kl + wave + terrain	0.79	2.11	1.35
rh_distance + wave_distance + wave_kl + wave + terrain	0.78	2.13	1.37
rh_distance + terrain	0.78	2.12	1.35
rh_distance + wave + terrain	0.78	2.12	1.36
rh_distance + wave_distance + wave + terrain	0.78	2.14	1.36
rh_distance + wave_distance + wave_kl + terrain	0.78	2.19	1.42
rh_distance + wave_distance + terrain	0.78	2.20	1.42
wave_distance + wave_kl + wave + terrain	0.64	3.02	2.16
wave_kl + wave + terrain	0.63	2.98	2.07
wave_distance + wave_kl + wave	0.63	3.07	2.16
wave_distance + wave_kl	0.62	3.23	2.35
wave_kl	0.62	3.19	2.31
wave_distance + wave	0.62	3.03	2.16
wave_distance + wave + terrain	0.62	3.05	2.13
wave_distance + wave_kl + terrain	0.62	3.22	2.30
wave_distance	0.60	3.29	2.40
wave_distance + terrain	0.59	3.30	2.37
wave_kl + terrain	0.59	3.34	2.37
wave_kl + wave	0.59	3.16	2.19
wave + terrain	0.58	3.20	2.22
wave_pearson	0.52	3.44	2.39
terrain	0.51	3.54	2.52

Table 5.3: Grid search on all unique combinations of criteria for GEDICorrect

When simulating the waveform at the reported GEDI footprint location, a clear disagreement between both original and simulated waveforms can be seen, clearly showing a geolocation error in GEDI data (Figure 5.5). The same pattern is observed where, even after applying the `collocateWaves` method, the misalignment of both waveforms still persists. In contrast, after applying the combination `kl + rh_distance` criteria approach, this disagreement was significantly reduced, consequently resulting in a smaller difference in the RH95 metric (6.81 vs. 7.53 meters).

Moreover, to efficiently assess the combination of criteria, the grid search was performed in parallel using 16 processes, reducing the testing time to approximately 38 minutes per test. This significantly reduced the computational time comparing to what would have been required for sequential execution.

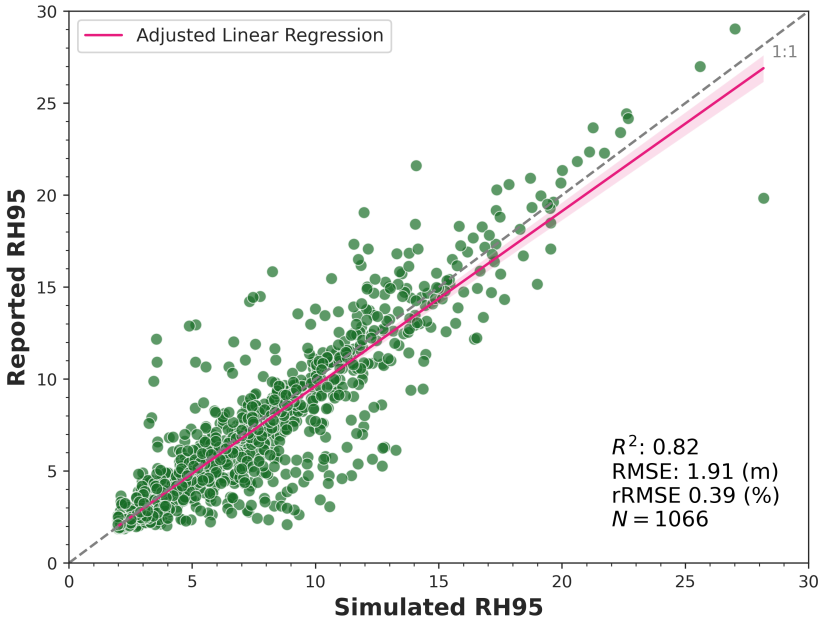


Figure 5.3: Relationship between reported (RH95_orb) and simulated (RH95_sim) GEDI canopy height using GEDICorrect with `kl+rh_distance` criteria

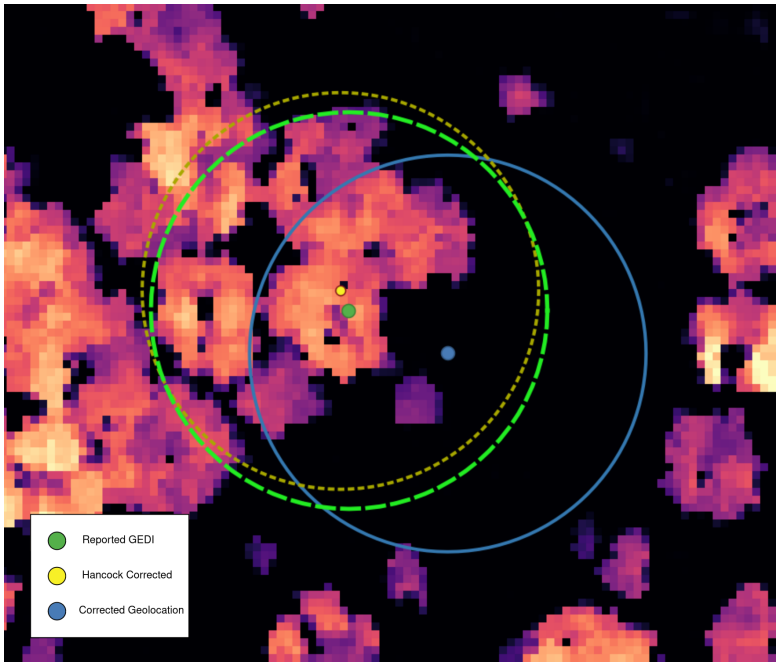


Figure 5.4: Illustration of the geolocation correction using `collocateWaves` and *GEDICorrect*'s `kl+rh_distance` criteria.

5.3 Efficiency Assessment

The framework enables parallel processing by creating a pool of processes that assigns blocks of footprints for correction to each process. This approach optimizes runtime by distributing the computational workload across multiple cores, leading to significant improvements in efficiency.

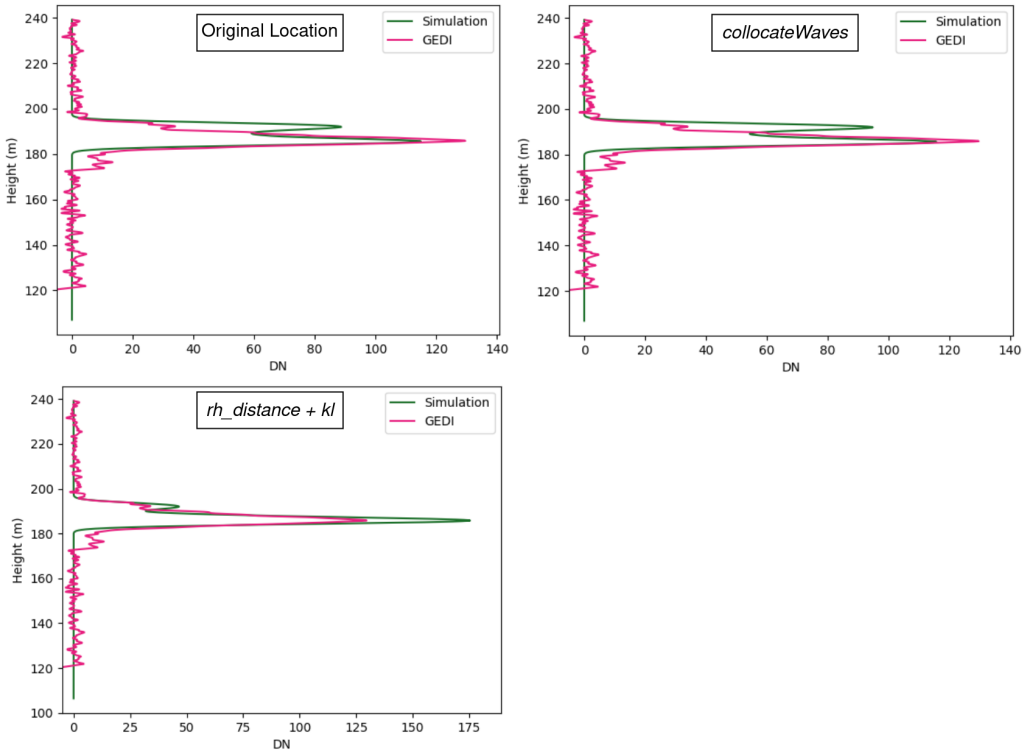


Figure 5.5: Waveforms of the reported and simulated at the original location and corrected geolocation footprints using `collocateWaves` and *GEDICorrect*'s `kl+rh_distance`. DN refers to the *Digital Number*, or the amplitude of the waveform.

A baseline test was performed where both GEDI Simulator and GEDICorrect were executed using a single process. Table 5.4 presents the real, user, and system times for these runs. The real time refers to the total elapsed time, while user time accounts for the actual CPU time spent on the user's code, and system time measures the time spent on system-level operations, such as I/O processing[1].

Test	N Processes	Real Time (min)	User Time (min)	Sys Time (min)
GEDI Simulator	1	5426.21	4736.58	646.28
GEDICorrect	1	258.51	216.5	42.1

Table 5.4: Baseline test comparison between GEDI Simulator and GEDICorrect

From the baseline results, it is evident that GEDICorrect is significantly more efficient than GEDI Simulator, despite both being run sequentially. The real time for GEDI Simulator is approximately $\simeq$ 90.5 hours (5426.21 minutes), compared to just over $\simeq$ 4 hours (258.51 minutes) for GEDICorrect. Additionally, the user and system time highlights GEDICorrect's more efficient use of computational resources and optimized I/O operations during the geolocation correction process, respectively.

To calculate the theoretical speedup gained from this method of parallelization, Amdahl's Law was used (Equation 4.5). Currently, GEDICorrect employs the parallelization strategy at the Simulation and Scoring steps (see section 4.5.1). To calculate S (Speedup, Equation 4.4), $P = 0.9$ (where P is the portion of the program that is parallelizable) was calculated by portioning 10% where the program is run sequentially.

Figure 5.6 demonstrates this theoretical speedup achieved by the program when system's resources are

[1] https://www.man7.org/linux/man-pages/man1/time.1.html

increased. In theory, GEDICorrect could achieve a maximum speedup of 10× at 1024 processes.

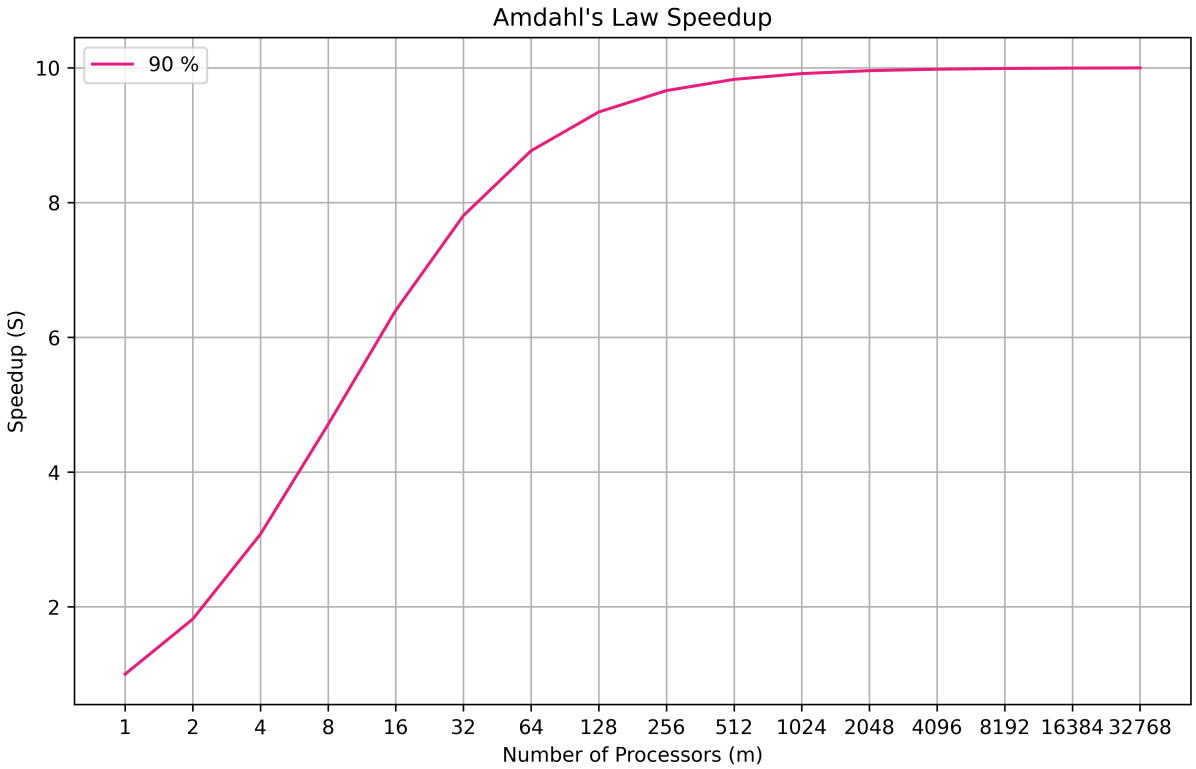


Figure 5.6: Amdahl's Law applied to GEDICorrect

Having established the baseline test and theoretical speedup, further experiments were conducted to evaluate the parallelization capacity of GEDICorrect and select the optimal number of processes. The number of processes was varied in powers of 2, starting from 2 and increasing up to the infrastructure (see section 4.3) limit of 32 cores, as shown in Table 5.5. Additionally, a test with 64 processes was included to observe whether increasing the number of processes beyond the available cores could yield further improvements or result in performance saturation.

p Processes	Real Time (min)	User Time (min)	Sys Time (min)	Speedup
2	145.11	223.36	44.45	1.78
4	86.2	231.51	44.8	2.99
8	55.57	240.40	44.51	4.64
16	44.90	278.22	50.19	5.75
32	37.59	328.28	62.03	6.86
64	36.7	325.7	57.28	7.03

Table 5.5: Test search on optimal number of processes to use for parallelization processing of GEDICorrect

Increasing the number of processes consistently reduced the real time running GEDICorrect up to 32 processes, at which point the runtime reduction began to plateau beyond the point of 64 processes, which indicated that the limit of parallel efficiency had been reached for the hardware infrastructure. This saturation point is common in parallel processing, where the overhead associated with managing additional processes begins to outweigh the benefits of parallelization, especially when the number of processes exceeds the number of physical cores [Rauber and Runger, 2010]. This is further reflected in the increased system time starting at 16 processes, which is higher than the elapsed real time.

In practice, GEDICorrect's speedup approaches the theoretical speedup previsions from Amdahl's Law (see Figure 5.6), which could be attributed to the high efficiency of its parallelization, as well as the framework's optimized memory management during the serial execution of the program.

5.4 Simulated Points Assessment

The geolocation correction process in GEDICorrect operates at the footprint level by simulating a set number of points around each footprint, distributed up to 12.5 meters from the original footprint centroid, with a minimum spacing of 1 meter between points. The purpose of this assessment is to identify the optimal number of simulated points in terms of accuracy and computational cost.

Table 5.6 presents the results for different numbers of simulated points, starting at 100, up to 300 in increases of 50.

N points	R^2	RMSE	MAE	Elapsed Real Time (min)
100	0.82	1.90	1.20	37.59
150	0.84	1.81	1.12	48.05
200	0.83	1.82	1.13	59.35
250	0.84	1.79	1.12	72.87
300	0.84	1.80	1.10	94.96

Table 5.6: Test search on number of simulated points around each footprint for geolocation correction with GEDICorrect

The results demonstrate that increasing the number of simulated points from 100 to 300 results in slight improvements in accuracy. The highest R^2 value is achieved with 150 points ($R^2 = 0.84$), while using further points does not lead to significant improvements but a rise in computation time is noticeable; for example, the time to process 300 points for each footprint is nearly 2.5 times longer than for 100 points, yet the increase in accuracy is relatively small. Figure 5.7 illustrates the spatial distribution of the simulated points for different values of *N*, as outlined in Table 5.6.

5.5 Stochasticity Assessment

The final experiment aims to assess the impact of randomness in the distribution of simulated points around each reported GEDI footprint and evaluate the reliability of GEDICorrect's footprint-level correction method. For this, an example footprint with shot number '44230300200152148' was selected for the study. Five separate simulations were conducted for this footprint, with each simulation introducing random variability in the location of generated points. Figures 5.8 and 5.9 illustrate the distribution of generated points for each of the five simulations, and the highest scored footprint (footprint with highest similarity with the reported waveform), respectively.

Table 5.7 presents the results of these tests, including the RH95 values for the optimal geolocation produced by each simulation and the final score obtained from Unit 3 (see Section 3.2.1) The Final Score column was calculated with respect to the combination of the `rh_distance` and `kl` criteria, ranging from 0 to 1, with 1 being the highest similarity a simulation can have in comparison to the reported footprint. The reported GEDI footprint recorded an RH95 of 6.80 meters, whereas the `collocateWaves` resulted in an RH95 of 8.06 meters, demonstrating the effectiveness of GEDICorrect's geolocation correction method. Figure 5.10 compares the simulated waveforms from each of the five simulation with the reported GEDI waveform, alongside the waveform generated from GEDI Simulator's optimal geolocation for the same footprint.

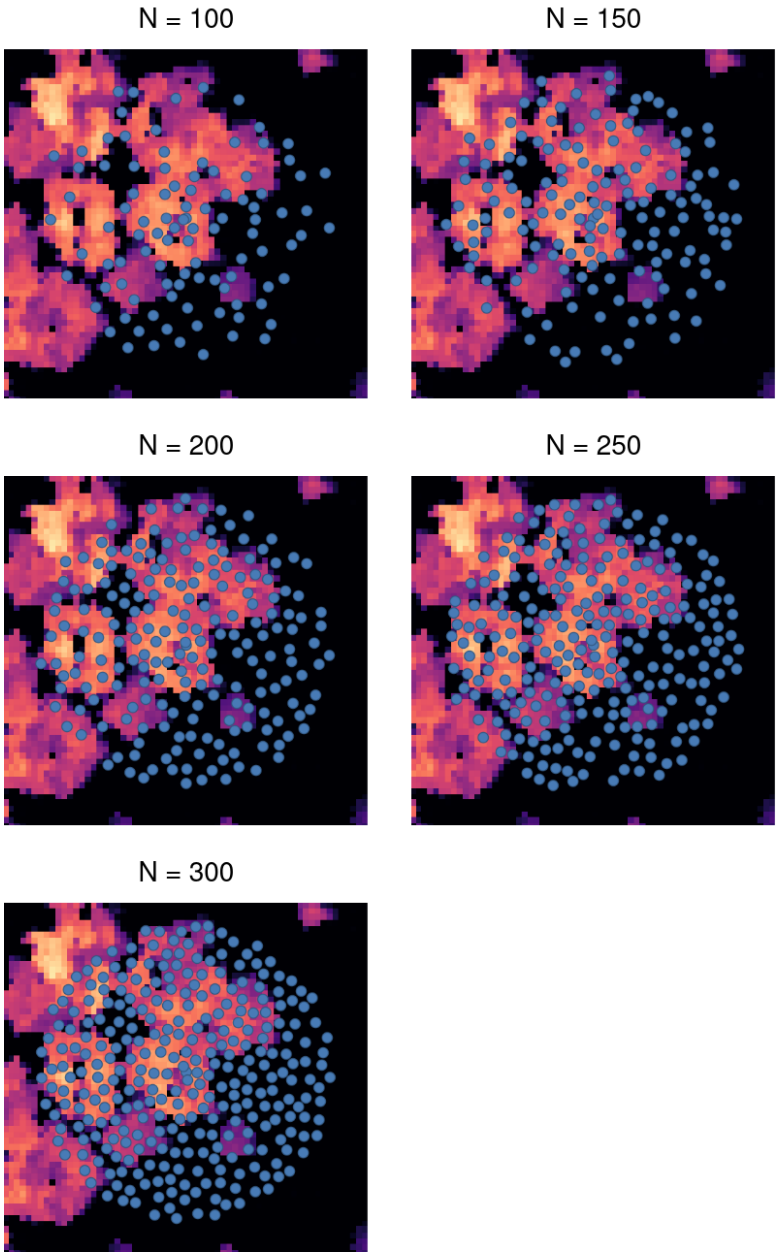


Figure 5.7: Distribution of simulated points across different simulations where N is changed.

Test	Simulated RH95	Final Score
1	7.22	0.96
2	6.98	0.96
3	6.67	0.94
4	6.82	0.94
5	7.75	0.95

Table 5.7: Stochasticity Assessment encompassing 5 different simulations

The results demonstrate that GEDICorrect successfully generates optimal points for footprint correction in most cases but can still leave small positional gaps. These gaps could be reduced by increasing the number of simulated points around each footprint, though this would come with a performance cost, as highlighted

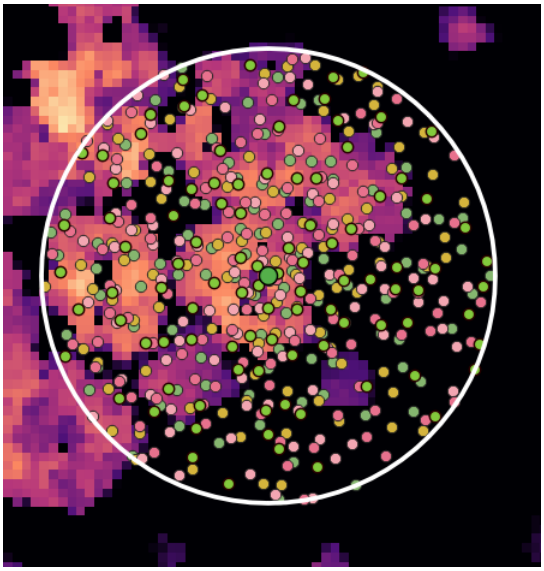


Figure 5.8: Generated footprint centroid points around reported footprints that span 5 different simulations using $N = 100$.

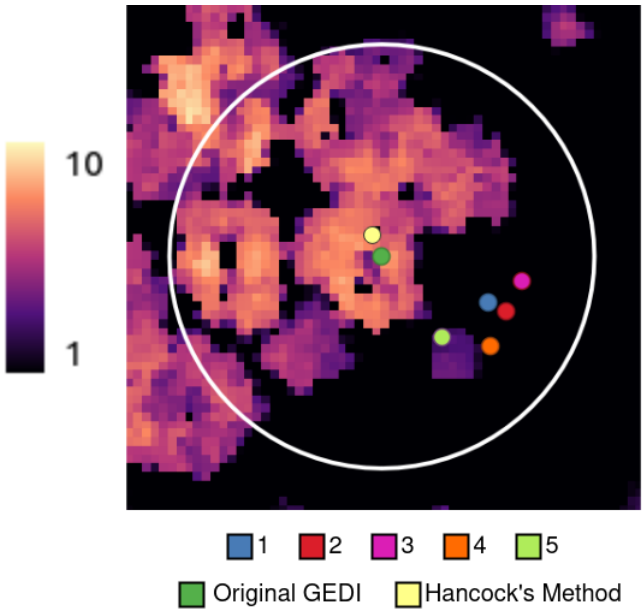


Figure 5.9: Five different highest scored simulations for reported footprint (shot_number: 44230300200152148) and for `collocateWaves` (Hancock's Method).

in Section 5.4 (see also Table 5.6). On average, the corrected footprints differ in distance by approximately 2 meters between simulations. The waveform plots in Figure 5.10 reveal a consistent distribution of energy across simulations, except near the canopy top (around 195 meters), where the energy levels vary slightly, which can be explained by the observed discrepancies in footprint location.Overall, a careful balance must be maintained between accuracy and computational efficiency, particularly when increasing the number of generated points around each reported footprint.

Moreover, it is possible to state that GEDICorrect consistently improves geolocation accuracy at the footprint level, particularly in areas with varying canopy heights. By leveraging parallel processing, efficient

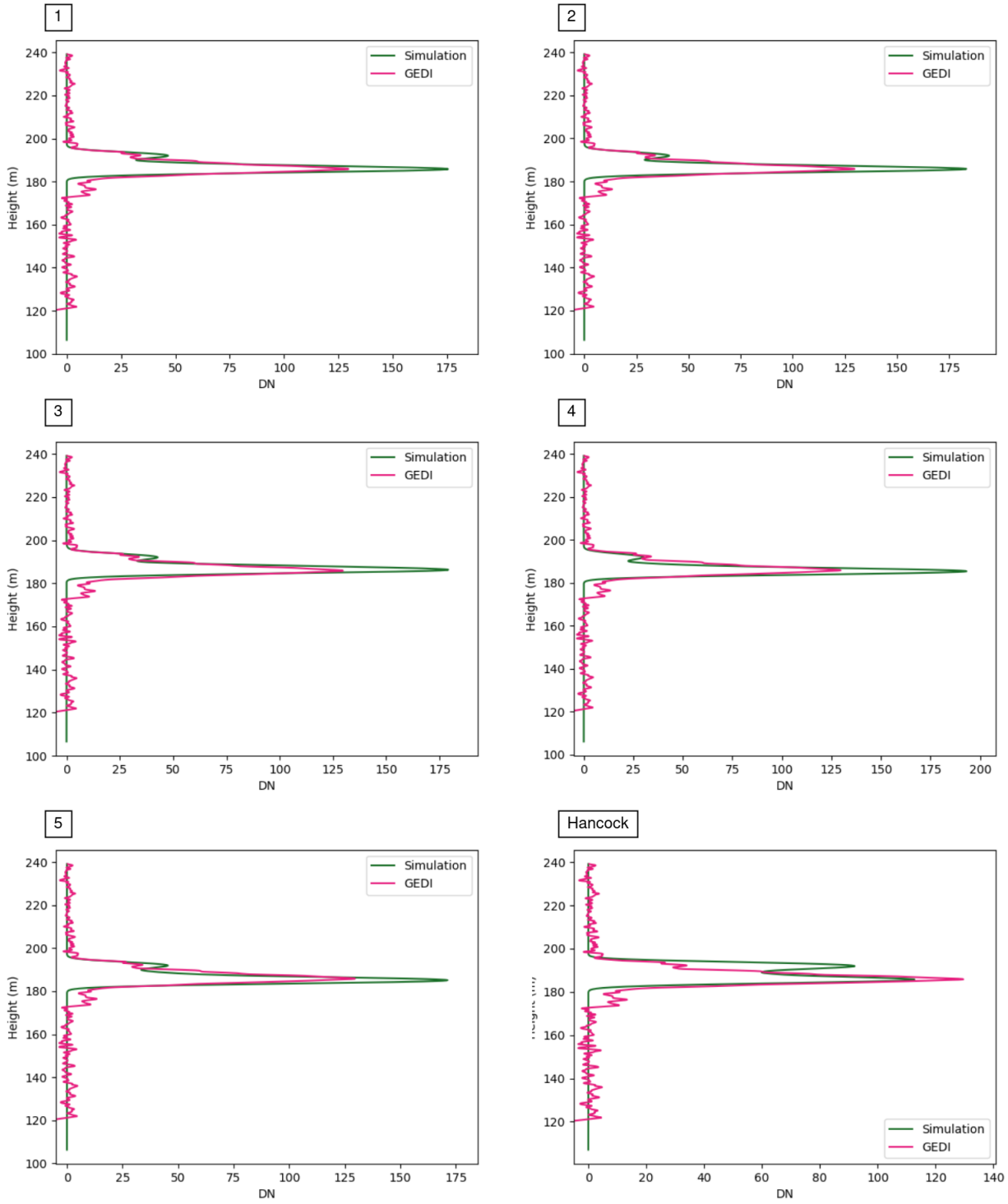


Figure 5.10: Waveforms plots for each of the different simulations (1 through 5), and for `collocateWaves` (Hancock) optimal position.

memory management, and introducing new geolocation correction criteria, GEDICorrect proves to be a faster, scalable and viable solution for improving GEDI geolocation error in complex landscapes.

6

Conclusion

This dissertation presents a comprehensive approach to improve the geolocation accuracy of GEDI footprints through the development of the *GEDICorrect* framework; it introduces new geolocation correction criteria, leverages parallel processing, and addresses critical limitations of existing methods. The key findings from this work demonstrate the following significant advancements:

1. **Enhanced Geolocation Accuracy**: By comparing *GEDICorrect* to the standard GEDI Simulator's collocateWaves method, it is evident that *GEDICorrect* significantly improves geolocation accuracy, particularly through the integration of the newly introduced criteria such as rh_distance and the combination of kl and rh_distance. These criteria more effectively capture both the waveform similarity and RH profile differences, which are critical indicators of geolocation accuracy. The results showed an R^2 improvement from 0.52 with *collocateWaves* to 0.86 using RH profile differences in *GEDICorrect*, highlighting the superiority of this approach.

2. **Efficient Computational Performance**: One of the most significant contributions of *GEDICorrect* is its efficient parallel processing design, which dramatically reduces computational time. This parallelization strategy leads to a speedup of 20x compared to the standard sequential method, making

large scale geolocation correction feasible. By optimizing memory resources utilization and reducing runtime from days to hours, *GEDICorrect* enhances scalability for processing large scale GEDI datasets.

3. **Pioneering Correction Criteria**: The introduction and successful application of novel criteria, particularly the rh_distance metric, but also the KL index, underscore the innovative nature of this research. By showing that the RH profile is a reliable indicator for geolocation accuracy, the work here presented opens new avenues for further refinement of waveform-based geolocation correction strategies in future studies.

Overall, the findings of this dissertation contribute to the ongoing efforts to improve the accuracy of spaceborne lidar measurements, in particular by paving the way for more reliable GEDI-derived metrics, which will have positive implications in forest monitoring, biodiversity assessments, and carbon accounting.

6.1 Future Work

While *GEDICorrect* has shown promising results, there are several areas that can be explored to further enhance its functionality and efficiency, some of them are worth mentioning:

- **Containerization** - containerizing the system using Docker or similar technologies would simplify the deployment process, making it easier to transfer and run the framework and required dependencies on various platforms without compatibility issues;
- **Adaptive Criteria Weighting** - currently, all criteria in *GEDICorrect* contribute equally to the scoring mechanism. Future enhancements could explore adaptive weighting schemes that prioritize certain criteria based on study objectives, or even employ Machine Learning to dynamically adjust these weights to optimize geolocation correction performance across various landscapes;
- **GPU Processing** - leveraging GPU processing to accelerate computations could significantly reduce runtime. However, this might require substantial adaptation of the current system, especially considering the limitations of GEDI Simulator, which may not natively support GPU acceleration;
- **Wall-to-wall Canopy Height Mapping** - since *GEDICorrect* improves the accuracy of GEDI geolocation, future research could focus on assessing how these corrections impact the generation of wall-to-wall canopy height maps (which are spatially comprehensive maps of canopy height that cover an entire region) by comparing them with ALS-derived canopy height models (CHM).

A

Developed Software

A.1 GEDI Simulator Installation Script

```
#!/bin/bash -f

HOMDIR="$HOME"

# set up environment variables
export ARCH=`uname -m`
export PATH=$PATH:./:$HOMDIR/bin/$ARCH:$HOMDIR/bin/csh
export GEDIRAT_ROOT=$HOMDIR/src/gedisimulator
export CMPFIT_ROOT=$HOMDIR/src/cmpfit-1.2
export GSL_ROOT=/usr/local/lib
export LIBCLIDAR_ROOT=$HOMDIR/src/libclidar
export HANCOCKTOOLS_ROOT=$HOMDIR/src/tools
export HDF5_LIB=/usr/lib/x86_64-linux-gnu
```

```
# Setup library paths
envFile="$HOMDIR/.bashrc"
echo "export ARCH=`uname -m`" >> $envFile
echo "export PATH=$PATH:./:$HOMDIR/bin/$ARCH:$HOMDIR/bin/csh" >> $envFile
echo "export GEDIRAT_ROOT=$HOMDIR/src/gedisimulator" >> $envFile
echo "export CMPFIT_ROOT=$HOMDIR/src/cmpfit-1.2" >> $envFile
echo "export GSL_ROOT=/usr/local/lib" >> $envFile
echo "export LIBCLIDAR_ROOT=$HOMDIR/src/libclidar" >> $envFile
echo "export HANCOCKTOOLS_ROOT=$HOMDIR/src/tools" >> $envFile
echo "export HDF5_LIB=/usr/lib/x86_64-linux-gnu" >> $envFile

# set up directory structure
if [ ! -e $HOMDIR/src ];then
  mkdir $HOMDIR/src
fi
if [ ! -e $HOMDIR/bin ];then
  mkdir $HOMDIR/bin
fi
if [ ! -e $HOMDIR/bin/$ARCH ];then
  mkdir $HOMDIR/bin/$ARCH
fi
if [ ! -e $HOMDIR/bin/csh ];then
  mkdir $HOMDIR/bin/csh
fi

# Install CMPFIT
pushd $HOMDIR/src
wget https://www.physics.wisc.edu/~craigm/idl/down/cmpfit-1.2.tar.gz
tar -xvf cmpfit-1.2.tar.gz
popd

pushd $HOMDIR/src
git clone https://bitbucket.org/StevenHancock/libclidar
git clone https://bitbucket.org/StevenHancock/tools
git clone https://bitbucket.org/StevenHancock/gedisimulator


programList="gediRat gediMetric mapLidar collocateWaves lasPoints fitTXpulse"
cd $GEDIRAT_ROOT/
make clean

for program in $programList;do
  make THIS=$program
  make THIS=$program install
done

programList="gediRatList.csh listGediWaves.csh overlapLasFiles.csh filtForR.csh"
for program in $cshList;do
```

```
    cp $program $HOMDIR/bin/csh/
  done

  popd
```

A.2 Dataset L1B-L2A Merging Script

```
import geopandas as gpd
import os
import pandas as pd

l1b_dir = "/home/yoru/personal/GEDI-Pipeline/GEDI-Pipeline/
    FUELSAT_TEST_L1B"
l2a_dir = "/home/yoru/personal/GEDI-Pipeline/GEDI-Pipeline/
    FUELSAT_TEST_L2A"
out_dir = "./MERGED_L1B_L2A"

l1b_files = [f for f in os.listdir(l1b_dir) if f.endswith(".gpkg")]
l2a_files = [f for f in os.listdir(l2a_dir) if f.endswith(".gpkg")]

l2a_dict = {}

# Align each L1B with L2A filenames
for l1b in l1b_files:
    l1b_ext = l1b.split("_")[2]

    for l2a in l2a_files:
        filename_ext = l2a.split("_")[2]

        if filename_ext == l1b_ext:
            l2a_dict[l1b] = l2a
            break

for file in l1b_files:
    l1b_file = gpd.read_file(os.path.join(l1b_dir, file), engine='pyogrio
        ')
    l2a_file = gpd.read_file(os.path.join(l2a_dir, l2a_dict[file]),
        engine='pyogrio')

    # L2A variables to keep
    cols_to_keep = ['shot_number', 'degrade_flag', 'quality_flag', '
        elev_lowestmode', 'digital_elevation_model',
                    'num_detectedmodes', 'solar_elevation', 'sensitivity'
                        ]

    cols_to_keep = cols_to_keep + [f"rh_{i}" for i in range(1, 101)]
```

```
34
35      rename_col = {}
36      for i in range(1,101):
37          rename_col[f'rh_{i}'] = f'rh_{i}'
38      rename_col['shot_number_x'] = 'shot_number'
39
40      # Introduce L2A data columns to final_df
41      l2a_file_to_merge = l2a_file[cols_to_keep]
42      l2a_file_to_merge = l2a_file_to_merge.rename(columns=rename_col)
43
44      # Align both data products by each footprint shot_number (since they
            are unique)
45      merged_df = pd.merge(l1b_file, l2a_file_to_merge, left_on='
            shot_number_x', right_on=[sn for sn in l2a_file_to_merge.columns
            if sn.endswith('shot_number')][0])
46
47      # Quality flags
48      merged_df = merged_df.query('degrade_flag == 0 and quality_flag == 1'
            )
49
50      merged_df = merged_df.query('sensitivity < 0.9')
51
52      merged_df = merged_df.query('solar_elevation < 0')
53
54      merged_df = merged_df.query('rh_95 <= 30')
55
56      merged_df = merged_df.query('rh_95 > 10 and num_detectedmodes == 1')
57
58      final_df = merged_df.loc[~((merged_df['rh_98'] > 5) & (merged_df['
            num_detectedmodes'] == 1))]
59
60      if len(final_df) < 1:
61          print(f"Filtered merged df {file} is empty, skipping")
62          continue
63
64      print("Saving ", file)
65      final_df.to_file(os.path.join(out_dir, file), driver="GPKG")
```

Listing A.1: Script developed to merge both GEDI L1B and L2A as well as the filtering process

A.3 GEDICorrect Simulation Unit

```
1 """
2 Handles the simulation of points around footprints and contains functions
      for processing the C program of gediSimulator
3 """
```

```
4  import os
5
6  import multiprocessing
7  import numpy as np
8  import geopandas as gpd
9  import pandas as pd
10
11 import laspy
12 from shapely.geometry import box, Point
13
14 import subprocess
15 from .data_process import parse_txt, parse_simulated_h5
16
17 def init_random_seed():
18     '''
19     Initializes a random seed for each multiprocessing process. This
           works to ensure
20     that no other worker process shares the inherited seed from the
           parent process
21     '''
22     seed = multiprocessing.current_process().pid  # Use process ID as the
           seed
23     np.random.seed(seed)
24
25
26 def generate_random_points(centroid_x, centroid_y, num_points, max_radius
       =12.5, min_dist=1.0):
27     '''
28     Generates a random number of ***num_points*** points around (x,y)
           coordinates up to a
29     ***max_radius*** distance, at ***min_dist*** intervals between
           generated points.
30     '''
31     centroid = Point(centroid_x, centroid_y)
32
33     # Define the boundary of the circle within which points will be
           placed
34     boundary = centroid.buffer(max_radius)
35     points = []
36
37     # Keep trying until all simulated points are valid
38     while len(points) < num_points:
39         angle = np.random.uniform(0, 2 * np.pi)
40         distance = np.random.uniform(0, max_radius)
41         x = centroid_x + np.cos(angle) * distance
42         y = centroid_y + np.sin(angle) * distance
43         new_point = Point(x, y)
44
```

```
45          # Check if the new point is at least min_dist meters away from
                all other points
46          if all(new_point.distance(other) >= min_dist for other in points)
                :
47              points.append(new_point)
48
49          # If the points list is filled and all are valid, exit the loop
50          if len(points) == num_points:
51              break
52
53      return points
54
55
56  def process_footprint(footprint, temp_dir, original_df, crs,
        simulate_original=True, num_points=100, max_radius=12.5, min_dist=1.0)
        :
57      '''
58      Core of GEDI Simulation footprints.
59      1 - Generates points around footprint centroid
60      2 - Runs the desired simulations from GediRat and GediMetrics from
            Steven Hancock
61      3 - Parses and processes the output of the simulations to be returned
62      '''
63
64      idx = multiprocessing.current_process().pid   # Get current process
            unique id
65
66      # Shot number
67      shot_number = footprint['shot_number_x']
68      original_fpt = original_df.loc[original_df['shot_number_x'] ==
            shot_number]
69
70      # Nbins
71      nbins = str(original_fpt['rx_sample_count'].values[0]+1)
72
73      ## Generate random points around footprint
74      rand_points = generate_random_points(footprint['geometry'].x,
            footprint['geometry'].y, num_points=num_points, max_radius=
            max_radius, min_dist=min_dist)
75
76      ## Generate txt list of coordinates from random point
77      with open(os.path.join(temp_dir, f"points_test_{idx}.txt"), "w") as f
            :
78          if simulate_original:
79              f.write(f"{footprint['geometry'].x} {footprint['geometry'].y
                    }\n") # Write original footprint position as first point
80              num_points += 1  # Additional point
81          for point in rand_points:
82              f.write(f"{point.x} {point.y}\n")
```

```
83
84      h5_file_dir = os.path.join(temp_dir, f"simu_wavef_{idx}.h5")
85      points_file_dir = os.path.join(temp_dir, f"points_test_{idx}.txt")
86      metric_outroot = os.path.join(temp_dir, f"{idx}_")
87
88      ## Simulate waveforms
89      exit_code = subprocess.run(["gediRat", "-inList", os.path.join(
            temp_dir, "alsList.txt"), "-listCoord", points_file_dir, "-hdf", "
            -aEPSG", "32629", "-ground", "-maxBins", nbins, "-output",
            h5_file_dir], stdout=subprocess.DEVNULL)
90      exit_code = subprocess.run(["gediMetric", "-input", h5_file_dir, "-
            readHDFgedi", "-ground", "-varScale", "3.5", "-sWidth", "0.8", "-
            rhRes", "1", "-laiRes", "5", "-outRoot", metric_outroot], stdout=
            subprocess.DEVNULL)
91
92      ## Handle each output
93      txt_df = parse_txt(footprint['shot_number_x'], metric_outroot+'.
            metric.txt') ######## TODO: Transform shotnumber to string and csv
             must display differently
94
95      try:
96          h5_df  = parse_simulated_h5(h5_file_dir, num_points)
97      except ValueError as e:
98          return []
99
100     # Concat the TXT and H5 dataframes
101     all_df = pd.concat([txt_df, h5_df], axis=1)
102
103     # Filter out NaN and add Geometry
104     all_df = all_df.dropna(axis=0)
105     all_df['geometry'] = list(zip(all_df.lon, all_df.lat))
106     all_df['geometry'] = all_df['geometry'].apply(Point)
107
108     # Filter out special case footprints
109     if len(all_df) < num_points:
110         # Did not simulate all points, discard
111         return []
112
113     # Sanity check: Check if vegetation was cut with original rh95
114     original_rh95 = original_fpt['rh_95'].values[0]
115     rh95_simulated_position = all_df['rhGauss_95']
116
117     # If mean difference between RH95 of Simulated and GEDI
118     mean_diffrh95 = (rh95_simulated_position - original_rh95).mean()
119     if mean_diffrh95 < -10:
120         # If negative, possibly a vegetation cut and datum difference
                between ALS and GEDI
121         return [shot_number]
122
```

```
123     point_df = gpd.GeoDataFrame(all_df, geometry='geometry')
124 
125     ## Return corrected footprint
126     return point_df
```

Listing A.2: Simulation Unit functions

A.4 GEDICorrect Execution Script

```
1  import os
2  import argparse
3  
4  from src.correct import GEDICorrect
5  
6  # -------------------------COMMAND LINE ARGUMENTS AND ERROR HANDLING
       ---------------------------- #
7  # Set up argument and error handling
8  parser = argparse.ArgumentParser(description='A script to correct GEDI
       Geolocation at the footprint level.')
9  
10 parser.add_argument('--granules_dir', required=False, help='Local
       directory where all GEDI files ', type=str)
11 
12 parser.add_argument('--input_file', required=False, help='GEDI File to be
        processed and corrected', type=str)
13 
14 parser.add_argument('--las_dir', required=True, help='Directory of .LAS
       files required for processing. Must intersect with input granule file(
       s)', type=str)
15 
16 parser.add_argument('--out_dir', required=True, help='Directory in which
       to save the corrected input granules and simulated points', type=str)
17 
18 parser.add_argument('--save_sim_points', required=False, help='Option to
       save all the simulated points around each footprint from the input
       data.',
19                     action='store_true')
20 
21 parser.add_argument('--save_origin_location', required=False, help='
       Option to save all the simulated reported locations for each footprint
        from the input data.',
22                     action='store_true')
23 
24 parser.add_argument('--criteria', required=True, help='Set of criteria to
        select the best footprint. Select from "wave", "rh", "rh_correlation"
        and "terrain". \
```

```
25                                                                  Select "all" to
                                                                       evaluate all
                                                                       simulated
                                                                       footprints with
                                                                        all the
                                                                       possible
                                                                       criteria', type
                                                                       =str, default='
                                                                       all')
26
27 parser.add_argument('--n_points', required=False, help='Number of points
       to simulated around each input footprint', type=int, default=100)
28
29 parser.add_argument('--radius', required=False, help='Maximum value for
       radius to simulate points around each original footprint', type=float,
        default=12.5)
30
31 parser.add_argument('--min_dist', required=False, help='Minimum distance
       between simulated points around each original footprint', type=float,
       default=1.0)
32
33 parser.add_argument('--rh_match', required=False, help='Relative Height
       metric to which make the Matching. Must require using "rh" criterion.'
       , type=int, default=95)
34
35 parser.add_argument('--parallel', required=False, help='Use parallel
       processing with "--n_processes" processes. If no n_processes are
       defined, defaults to all minus 2 processes.',
36                      action='store_true')
37
38 parser.add_argument('--n_processes', required=False, help='Number of
       processes to use for parallel processing. If none are specified,
       defaults to all minus 2 processes.', type=int)
39
40 args = parser.parse_args()
41
42 # List Files and Create Output directory if needed
43
44 if not os.path.exists(args.out_dir):
45     os.mkdir(args.out_dir)
46
47 input_granules = None
48
49 if args.granules_dir:
50     input_granules = [os.path.join(args.granules_dir, f) for f in os.
          listdir(args.granules_dir) if f.endswith('.gpkg')]
51
52 if args.input_file:
53     input_granules = [args.input_file]
```

```
54
55 correct = GEDICorrect(granule_list=input_granules,
56                       las_dir=args.las_dir,
57                       out_dir=args.out_dir,
58                       criteria=args.criteria,
59                       rh=args.rh_match,
60                       save_sim_points=args.save_sim_points,
61                       save_origin_location=args.save_origin_location,
62                       use_parallel=args.parallel,
63                       n_processes=args.n_processes)
64
65 results = correct.simulate(args.n_points, args.radius, args.min_dist)
66
67 print(f"[Correction] Correction of input footprints complete! All files
      have been saved to {args.out_dir}")
```

Listing A.3: Script developed to execute the GEDICorrect framework

Bibliography

[Abdalati et al., 2010] Abdalati, W., Zwally, H. J., Bindschadler, R., Csatho, B., Farrell, S. L., Fricker, H. A., Harding, D., Kwok, R., Lefsky, M., Markus, T., Marshak, A., Neumann, T., Palm, S., Schutz, B., Smith, B., Spinhirne, J., and Webb, C. (2010). The icesat-2 laser altimetry mission. *Proceedings of the IEEE*, 98(5):735–751.

[Adam et al., 2020] Adam, M., Urbazaev, M., Dubois, C., and Schmullius, C. (2020). Accuracy assessment of gedi terrain elevation and canopy height estimates in european temperate forests: Influence of environmental and acquisition parameters. *Remote Sensing*, 12(23).

[Amdahl, 1967] Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, AFIPS '67 (Spring), page 483–485, New York, NY, USA. Association for Computing Machinery.

[Andrew, 1979] Andrew, A. (1979). Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219.

[Bachman, 1979] Bachman, C. G. (1979). Laser radar systems and techniques. *Dedham*.

[Barney and Frederick, 2024] Barney, B. and Frederick, D. (2024). Introduction to Parallel Computing Tutorial | HPC @ LLNL — hpc.llnl.gov. https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial. [Accessed 07-10-2024].

[Beck et al., 2021] Beck, J., Wirt, B., Armston, J., Hofton, M., Luthcke, S., and Tang, H. (2021). Global ecosystem dynamics investigation (gedi) level 02 user guide. document version, 2.

[Bergen et al., 2009] Bergen, K., Goetz, S., Dubayah, R., Henebry, G., Hunsaker, C., Imhoff, M., Nelson, R., Parker, G., Radeloff, V., and Bergen, C. (2009). Remote sensing of vegetation 3-d structure for biodiversity and habitat: Review and implications for lidar and radar spaceborne missions. *Journal of Geophysical Research*, 114.

[Blair and Hofton, 1999] Blair, J. B. and Hofton, M. A. (1999). Modeling laser altimeter return waveforms over complex vegetation using high-resolution elevation data. *Geophysical Research Letters*, 26(16):2509–2512.

[Carrasco et al., 2019] Carrasco, L., Giam, X., Papeș, M., and Sheldon, K. S. (2019). Metrics of lidar-derived 3d vegetation structure reveal contrasting effects of horizontal and vertical forest heterogeneity on bird species richness. *Remote Sensing*, 11(7).

[Chang, 2018] Chang, K.-T. (2018). *Introduction to geographic information systems*.

[Chapin et al., 2011] Chapin, F. S., Matson, P. A., and Vitousek, P. M. (2011). *Principles of Terrestrial Ecosystem Ecology*. Springer New York.

[Chunyu et al., 2017] Chunyu, Y., Kun, X., Yunfei, B., Nan, Z., and Hongyan, H. (2017). A matching method of space-borne laser altimeter big footprint waveform and terrain based on cross cumulative residual entropy. *Acta Geodaetica et Cartographica Sinica*, 46(3):346.

[Corado and Godinho, 2024] Corado, L. and Godinho, S. (2024). *leonelluiscorado/GEDI-Pipeline*.

[Cracknell, 2007] Cracknell, A. P. (2007). *Introduction to remote sensing*. CRC Press, Boca Raton, FL, 2 edition.

[Craig B. Markwardt, 2022] Craig B. Markwardt (2022). Cmpfit.

[de Conto et al., 2024] de Conto, T., Armston, J., and Dubayah, R. (2024). Characterizing the structural complexity of the earth's forests with spaceborne lidar. *Nature Communications*, 15(1).

[De Frenne et al., 2021] De Frenne, P., Lenoir, J., Luoto, M., Scheffers, B. R., Zellweger, F., Aalto, J., Ashcroft, M. B., Christiansen, D. M., Decocq, G., De Pauw, K., Govaert, S., Greiser, C., Gril, E., Hampe, A., Jucker, T., Klinges, D. H., Koelemeijer, I. A., Lembrechts, J. J., Marrec, R., Meeussen, C., Ogée, J., Tyystjärvi, V., Vangansbeke, P., and Hylander, K. (2021). Forest microclimates and climate change: Importance, drivers and future research agenda. *Global Change Biology*, 27(11):2279–2297.

[Dhargay et al., 2022] Dhargay, S., Lyell, C. S., Brown, T. P., Inbar, A., Sheridan, G. J., and Lane, P. N. J. (2022). Performance of gedi space-borne lidar for quantifying structural variation in the temperate forests of south-eastern australia. *Remote Sensing*, 14(15).

[Dong and Chen, 2017] Dong, P. and Chen, Q. (2017). *LiDAR Remote Sensing and Applications*. CRC Press.

[Dorado-Roda et al., 2021] Dorado-Roda, I., Pascual, A., Godinho, S., Silva, C. A., Botequim, B., Rodríguez-Gonzálvez, P., González-Ferreiro, E., and Guerra-Hernández, J. (2021). Assessing the accuracy of gedi data for canopy height and aboveground biomass estimates in mediterranean forests. *Remote Sensing*, 13(12).

[Drake et al., 2002] Drake, J. B., Dubayah, R. O., Knox, R. G., Clark, D. B., and Blair, J. (2002). Sensitivity of large-footprint lidar to canopy structure and biomass in a neotropical rainforest. *Remote Sensing of Environment*, 81(2):378–392.

[Dubayah et al., 2020] Dubayah, R., Blair, J. B., Goetz, S., Fatoyinbo, L., Hansen, M., Healey, S., Hofton, M., Hurtt, G., Kellner, J., Luthcke, S., Armston, J., Tang, H., Duncanson, L., Hancock, S., Jantz, P., Marselis, S., Patterson, P. L., Qi, W., and Silva, C. (2020). The global ecosystem dynamics investigation: High-resolution laser ranging of the earth's forests and topography. *Science of Remote Sensing*, 1:100002.

[Duncanson et al., 2020] Duncanson, L., Neuenschwander, A., Hancock, S., Thomas, N., Fatoyinbo, T., Simard, M., Silva, C. A., Armston, J., Luthcke, S. B., Hofton, M., Kellner, J. R., and Dubayah, R. (2020). Biomass estimation from simulated gedi, icesat-2 and nisar across environmental gradients in sonoma county, california. *Remote Sensing of Environment*, 242:111779.

[East et al., 2024] East, A., Hansen, A., Jantz, P., Currey, B., Roberts, D. W., and Armenteras, D. (2024). Validation and error minimization of global ecosystem dynamics investigation (gedi) relative height metrics in the amazon. *Remote Sensing*, 16(19).

[Fayad et al., 2021] Fayad, I., Baghdadi, N., Alcarde Alvares, C., Stape, J. L., Bailly, J. S., Scolforo, H. F., Cegatta, I. R., Zribi, M., and Le Maire, G. (2021). Terrain slope effect on forest height and wood volume estimation from gedi data. *Remote Sensing*, 13(11).

[Fayad et al., 2020] Fayad, I., Baghdadi, N., Bailly, J. S., Frappart, F., and Zribi, M. (2020). Analysis of gedi elevation data accuracy for inland waterbodies altimetry. *Remote Sensing*, 12(17).

[Fayad et al., 2022] Fayad, I., Baghdadi, N., and Lahssini, K. (2022). An assessment of the gedi lasers' capabilities in detecting canopy tops and their penetration in a densely vegetated, tropical area. *Remote Sensing*, 14(13).

[Filin, 2006] Filin, S. (2006). Calibration of spaceborne laser altimeters-an algorithm and the site selection problem. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1484–1492.

[Glantz and Slinker, 1991] Glantz, S. A. and Slinker, B. K. (1991). *Primer of applied regression and analysis of variance*. McGraw-Hill Professional, New York, NY, 2 edition.

[Gough, 2009] Gough, B., editor (2009). *GNU scientific library reference manual*. Network Theory, Bristol, England, 3 edition.

[Grama et al., 2003] Grama, A., Karypis, G., Kumar, V., and Gupta, A. (2003). *Introduction to parallel computing*. Addison Wesley, Boston, MA, 2 edition.

[Gropp and Smith, 1990] Gropp, W. D. and Smith, E. B. (1990). Computational fluid dynamics on parallel processors. *Computers & Fluids*, 18(3):289–304.

[Guo et al., 2021] Guo, Q., Su, Y., Hu, T., Guan, H., Jin, S., Zhang, J., Zhao, X., Xu, K., Wei, D., Kelly, M., and Coops, N. C. (2021). Lidar boosts 3d ecological observations and modelings: A review and perspective. *IEEE Geoscience and Remote Sensing Magazine*, 9(1):232–257.

[Gwenzi et al., 2016] Gwenzi, D., Lefsky, M. A., Suchdeo, V. P., and Harding, D. J. (2016). Prospects of the icesat-2 laser altimetry mission for savanna ecosystem structural studies based on airborne simulation data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 118:68–82.

[Hall et al., 2011] Hall, F. G., Bergen, K., Blair, J. B., Dubayah, R., Houghton, R., Hurtt, G., Kellndorfer, J., Lefsky, M., Ranson, J., Saatchi, S., Shugart, H., and Wickland, D. (2011). Characterizing 3d vegetation structure from space: Mission requirements. *Remote Sensing of Environment*, 115(11):2753–2775. DESDynI VEG-3D Special Issue.

[Hancock et al., 2019] Hancock, S., Armston, J., Hofton, M., Sun, X., Tang, H., Duncanson, L. I., Kellner, J. R., and Dubayah, R. (2019). The gedi simulator: A large-footprint waveform lidar simulator for calibration and validation of spaceborne missions. *Earth and Space Science*, 6(2):294–310.

[Harding and Carabajal, 2005] Harding, D. J. and Carabajal, C. C. (2005). Icesat waveform measurements of within-footprint topographic relief and vegetation vertical structure. *Geophysical Research Letters*, 32(21).

[HDFGroup, 1987] HDFGroup (1987). Hdf5.

[Herlihy and Shavit, 2008] Herlihy, M. and Shavit, N. (2008). *The art of multiprocessor programming*. Morgan Kaufmann, Oxford, England.

[Hofton et al., 2019] Hofton, M., Blair, J., Story, S., and Yi, D. (2019). Algorithm theoretical basis document (atbd) for gedi transmit and receive waveform processing for l1 and l2 products. *University of Maryland: College Park, MD, USA*, 44.

[Hyndman and Koehler, 2006] Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

[Irish and White, 1998] Irish, J. and White, T. (1998). Coastal engineering applications of high-resolution lidar bathymetry. *Coastal Engineering*, 35(1):47–71.

[Jacob et al., 2012] Jacob, R., Krishna, J., Xu, X., Mickelson, S., Tautges, T., Wilde, M., Latham, R., Foster, I., Ross, R., Hereld, M., Larson, J., Bochev, P., Peterson, K., Taylor, M., Schuchardt, K., Yin, J., Middleton, D., Haley, M., Brown, D., Huang, W., Shea, D., Brownrigg, R., Vertenstein, M., Ma, K.-L., and Xie, J. (2012). Poster: Bringing task and data parallelism to analysis of climate model output. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pages 1495–1495.

[Jia et al., 2024] Jia, D., Wang, C., Hakkenberg, C. R., Numata, I., Elmore, A. J., and Cochrane, M. A. (2024). Accuracy evaluation and effect factor analysis of gedi aboveground biomass product for temperate forests in the conterminous united states. *GIScience & Remote Sensing*, 61(1):2292374.

[Jiang et al., 2023] Jiang, Y., Liu, B., Wang, R., Li, Z., Chen, Z., Zhao, B., Guo, G., Fan, W., Huang, F., and Yang, Y. (2023). Photon counting lidar working in daylight. *Optics & Laser Technology*, 163:109374.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

[Kyker-Snowman et al., 2021] Kyker-Snowman, E., Lombardozzi, D. L., Bonan, G. B., Cheng, S. J., Dukes, J. S., Frey, S. D., Jacobs, E. M., McNellis, R., Rady, J. M., Smith, N. G., Thomas, R. Q., Wieder, W. R., and Grandy, A. S. (2021). Increasing the spatial and temporal impact of ecological research: A roadmap for integrating a novel terrestrial process into an earth system model. *Global Change Biology*, 28(2):665–684.

[Lahssini et al., 2022] Lahssini, K., Baghdadi, N., le Maire, G., and Fayad, I. (2022). Influence of gedi acquisition and processing parameters on canopy height estimates over tropical forests. *Remote Sensing*, 14(24).

[Lechner et al., 2020] Lechner, A. M., Foody, G. M., and Boyd, D. S. (2020). Applications in remote sensing to forest ecology and management. *One Earth*, 2(5):405–412.

[Lefsky et al., 2002] Lefsky, M. A., Cohen, W. B., Parker, G. G., and Harding, D. J. (2002). Lidar Remote Sensing for Ecosystem Studies: Lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists. *BioScience*, 52(1):19–30.

[Lefsky et al., 2005] Lefsky, M. A., Harding, D. J., Keller, M., Cohen, W. B., Carabajal, C. C., Del Bom Espirito-Santo, F., Hunter, M. O., and de Oliveira Jr., R. (2005). Estimates of forest canopy height and aboveground biomass using icesat. *Geophysical Research Letters*, 32(22).

[Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

[Li et al., 2020] Li, S., Zhao, Y., Varma, R., Salpekar, O., Noordhuis, P., Li, T., Paszke, A., Smith, J., Vaughan, B., Damania, P., and Chintala, S. (2020). Pytorch distributed: Experiences on accelerating data parallel training.

[Li et al., 2023] Li, X., Wessels, K., Armston, J., Hancock, S., Mathieu, R., Main, R., Naidoo, L., Erasmus, B., and Scholes, R. (2023). First validation of gedi canopy heights in african savannas. *Remote Sensing of Environment*, 285:113402.

[Lillesand et al., 2015] Lillesand, T., Kiefer, R. W., and Chipman, J. (2015). *Remote sensing and image interpretation*. John Wiley & Sons, Nashville, TN, 7 edition.

[Lim et al., 2003] Lim, K., Treitz, P., Wulder, M., St-Onge, B., and Flood, M. (2003). Lidar remote sensing of forest structure. *Progress in Physical Geography: Earth and Environment*, 27(1):88–106.

[Luthcke et al., 2001] Luthcke, S., Carabajal, C., Rowlands, D., and Pavlis, D. (2001). Improvements in spaceborne laser altimeter data geolocation. *Surveys in Geophysics*, 22.

[Magruder et al., 2007] Magruder, L., Webb, C., Urban, T., Silverberg, E., and Schutz, B. (2007). Icesat altimetry data product verification at white sands space harbor. *Geoscience and Remote Sensing, IEEE Transactions on*, 45:147 – 155.

[Malambo and Popescu, 2024] Malambo, L. and Popescu, S. (2024). Mapping vegetation canopy height across the contiguous united states using icesat-2 and ancillary datasets. *Remote Sensing of Environment*, 309:114226.

[Mandlburger et al., 2019] Mandlburger, G., Lehner, H., and Pfeifer, N. (2019). A comparison of single photon and full waveform lidar. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5:397–404.

[Moudrý et al., 2022] Moudrý, V., Cord, A. F., Gábor, L., Laurin, G. V., Barták, V., Gdulová, K., Malavasi, M., Rocchini, D., Stereńczak, K., Prošek, J., Klápště, P., and Wild, J. (2022). Vegetation structure derived from airborne laser scanning to assess species distribution and habitat suitability: The way forward. *Diversity and Distributions*, 29(1):39–50.

[Neuenschwander et al., 2023] Neuenschwander, A., Pitts, K., Jelley, B., Jelley, Robbins, J., Markel, J., Popescu, S., Nelson, R., Harding, D., Pederson, Klotz, B., and Sheridan, R. (2023). Ice, cloud, and land elevation satellite (icesat-2) project algorithm theoretical basis document (atbd) for land - vegetation along-track products (atl08), version 6.

[Neumann et al., 2019] Neumann, T. A., Martino, A. J., Markus, T., Bae, S., Bock, M. R., Brenner, A. C., Brunt, K. M., Cavanaugh, J., Fernandes, S. T., Hancock, D. W., Harbeck, K., Lee, J., Kurtz, N. T., Luers, P. J., Luthcke, S. B., Magruder, L., Pennington, T. A., Ramos-Izquierdo, L., Rebold, T., Skoog, J., and Thomas, T. C. (2019). The ice, cloud, and land elevation satellite – 2 mission: A global geolocated photon product derived from the advanced topographic laser altimeter system. *Remote Sensing of Environment*, 233:111325.

[Niles Ritter, 2000] Niles Ritter (2000). libgeotiff.

[Olszewski, 2012] Olszewski, D. (2012). A probabilistic approach to fraud detection in telecommunications. *Knowledge-Based Systems*, 26:246–258.

[Pardini et al., 2019] Pardini, M., Dubayah, R., Fatoyinbo, L., Tello, M., Cazcarra-Bes, V., Choi, C., and Papathanassiou, K. (2019). Early lessons on combining lidar and multi-baseline sar measurements for forest structure characterization. *Surveys in Geophysics*, 40.

[Potapov et al., 2021] Potapov, P., Li, X., Hernandez-Serna, A., Tyukavina, A., Hansen, M. C., Kommareddy, A., Pickens, A., Turubanova, S., Tang, H., Silva, C. E., Armston, J., Dubayah, R., Blair, J. B., and Hofton, M. (2021). Mapping global forest canopy height through integration of gedi and landsat data. *Remote Sensing of Environment*, 253:112165.

[Quirós Rosado et al., 2021] Quirós Rosado, E., Polo, M.-E., and Fragoso Campón, L. (2021). Gedi elevation accuracy assessment: A case study of southwest spain. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP:1–1.

[Rauber and Runger, 2010] Rauber, T. and Runger, G. (2010). *Parallel Programming*. Springer, Berlin, Germany, 2010 edition.

[Rebonatto et al., 2017] Rebonatto, M. T., Schmitz, M. A., and Spalding, L. E. S. (2017). Methods of comparison and similarity scoring for electrical current waveforms. *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.

[Roman and Ursu, 2016] Roman, A. and Ursu, T. (2016). *Multispectral satellite imagery and airborne laser scanning techniques for the detection of archaeological vegetation marks*, pages 141–152.

[Rouault et al., 2024] Rouault, E., Warmerdam, F., Schwehr, K., Kiselev, A., Butler, H., Łoskot, M., Szekeres, T., Tourigny, E., Landa, M., Miara, I., Elliston, B., Chaitanya, K., Plesea, L., Morissette, D., Jolma, A., Dawson, N., Baston, D., de Stigter, C., and Miura, H. (2024). Gdal.

[Roy et al., 2021] Roy, D. P., Kashongwe, H. B., and Armston, J. (2021). The impact of geolocation uncertainty on gedi tropical forest canopy height estimation and change monitoring. *Science of Remote Sensing*, 4:100024.

[Ruoqi Wang and Li, 2024] Ruoqi Wang, Yagang Lu, D. L. and Li, G. (2024). Improving extraction of forest canopy height through reprocessing icesat-2 atlas and gedi data in sparsely forested plain regions. *GIScience & Remote Sensing*, 61(1):2396807.

[Saatchi et al., 2007] Saatchi, S., Halligan, K., Despain, D., and Crabtree, R. (2007). Estimation of forest fuel load from radar remote sensing. *Geoscience and Remote Sensing, IEEE Transactions on*, 45:1726 – 1740.

[Samvedan, 2020] Samvedan, S. (2020). Electromagnetic Spectrum — rsgislearn.blogspot.com. http://rsgislearn.blogspot.com/2007/04/electromagnetic-spectrum.html. [Accessed 30-09-2024].

[Schaller et al., 2024] Schaller, M., Borrow, J., Draper, P. W., Ivkovic, M., McAlpine, S., Vandenbroucke, B., Bahé, Y., Chaikin, E., Chalk, A. B. G., Chan, T. K., Correa, C., van Daalen, M., Elbers, W., Gonnet, P., Hausammann, L., Helly, J., Huško, F., Kegerreis, J. A., Nobels, F. S. J., Ploeckinger, S., Revaz, Y., Roper, W. J., Ruiz-Bonilla, S., Sandnes, T. D., Uyttenhove, Y., Willis, J. S., and Xiang, Z. (2024). Swift: a modern highly parallel gravity and smoothed particle hydrodynamics solver for astrophysical and cosmological applications. *Monthly Notices of the Royal Astronomical Society*, 530(2):2378–2419.

[Schott, 2007] Schott, J. R. (2007). *Remote Sensing: The Image Chain Approach*. Oxford University Press.

[Shannon et al., 2024] Shannon, E. S., Finley, A. O., Hayes, D. J., Noralez, S. N., Weiskittel, A. R., Cook, B. D., and Babcock, C. (2024). Quantifying and correcting geolocation error in spaceborne lidar forest canopy observations using high spatial accuracy data: A bayesian model approach. *Environmetrics*, 35(4):e2840.

[Shirkhorshidi et al., 2015] Shirkhorshidi, A. S., Aghabozorgi, S., and Wah, T. Y. (2015). A comparison study on similarity and dissimilarity measures in clustering continuous data. *PLOS ONE*, 10(12):1–20.

[Simard et al., 2011] Simard, M., Pinto, N., Fisher, J. B., and Baccini, A. (2011). Mapping forest canopy height globally with spaceborne lidar. *Journal of Geophysical Research: Biogeosciences*, 116(G4).

[Sirota et al., 2005] Sirota, J. M., Bae, S., Millar, P., Mostofi, D., Webb, C., Schutz, B., and Luthcke, S. (2005). The transmitter pointing determination in the geoscience laser altimeter system. *Geophysical Research Letters*, 32(22).

[Smith et al., 2019] Smith, B., Fricker, H. A., Holschuh, N., Gardner, A. S., Adusumilli, S., Brunt, K. M., Csatho, B., Harbeck, K., Huth, A., Neumann, T., Nilsson, J., and Siegfried, M. R. (2019). Land ice height-retrieval algorithm for nasa's icesat-2 photon-counting laser altimeter. *Remote Sensing of Environment*, 233:111352.

[Stysley et al., 2015] Stysley, P. R., Coyle, D. B., Kay, R. B., Frederickson, R., Poulios, D., Cory, K., and Clarke, G. (2015). Long term performance of the high output maximum efficiency resonator (homer) laser for nasa s global ecosystem dynamics investigation (gedi) lidar. *Optics and Laser Technology*, 68(Complete):67–72.

[Sumnall et al., 2016] Sumnall, M. J., Hill, R. A., and Hinsley, S. A. (2016). Comparison of small-footprint discrete return and full waveform airborne lidar data for estimating multiple forest variables. *Remote Sensing of Environment*, 173:214–223.

[Szpakowski and Jensen, 2019] Szpakowski, D. M. and Jensen, J. L. R. (2019). A review of the applications of remote sensing in fire ecology. *Remote Sensing*, 11(22).

[Tang et al., 2012] Tang, H., Dubayah, R., Swatantran, A., Hofton, M., Sheldon, S., Clark, D. B., and Blair, B. (2012). Retrieval of vertical lai profiles over tropical rain forests using waveform lidar at la selva, costa rica. *Remote Sensing of Environment*, 124:242–250.

[Tang et al., 2023] Tang, H., Stoker, J., Luthcke, S., Armston, J., Lee, K., Blair, B., and Hofton, M. (2023). Evaluating and mitigating the impact of systematic geolocation error on canopy height measurement performance of gedi. *Remote Sensing of Environment*, 291:113571.

[V.C. Oliveira et al., 2023] V.C. Oliveira, P., Zhang, X., Peterson, B., and Ometto, J. P. (2023). Using simulated gedi waveforms to evaluate the effects of beam sensitivity and terrain slope on gedi l2a relative height metrics over the brazilian amazon forest. *Science of Remote Sensing*, 7:100083.

[W. Wagner and Ducic, 2008] W. Wagner, M. Hollaus, C. B. and Ducic, V. (2008). 3d vegetation mapping using small-footprint full-waveform airborne laser scanners. *International Journal of Remote Sensing*, 29(5):1433–1452.

[Wang et al., 2022] Wang, C., Elmore, A. J., Numata, I., Cochrane, M. A., Shaogang, L., Huang, J., Zhao, Y., and Li, Y. (2022). Factors affecting relative height and ground elevation estimations of gedi among forest types across the conterminous usa. *GIScience & Remote Sensing*, 59(1):975–999.

[Wang et al., 2020] Wang, C., Yang, X., Xiaohuan, X., Zhang, H., Chen, S., Peng, S., and Zhu, X. (2020). Evaluation of footprint horizontal geolocation accuracy of spaceborne full-waveform lidar based on digital surface model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP:1–1.

[Wehr and Lohr, 1999] Wehr, A. and Lohr, U. (1999). Airborne laser scanning—an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):68–82.

[Xu et al., 2023] Xu, Y., Ding, S., Chen, P., Tang, H., Ren, H., and Huang, H. (2023). Horizontal geolocation error evaluation and correction on full-waveform lidar footprints via waveform matching. *Remote Sensing*, 15(3).

[Zellweger et al., 2019] Zellweger, F., De Frenne, P., Lenoir, J., Rocchini, D., and Coomes, D. (2019). Advances in microclimate ecology arising from remote sensing. *Trends in Ecology & Evolution*, 34(4):327–341.

[Zhao et al., 2022] Zhao, P., Li, S., Ma, Y., Liu, X., Yang, J., and Yu, D. (2022). A new terrain matching method for estimating laser pointing and ranging systematic biases for spaceborne photon-counting laser altimeters. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188:220–236.

[Zhou et al., 2016] Zhou, Y., Qiu, F., Ni, F., Lou, Y., Zhang, C., Alfarhan, M., and Al-Dosari, A. A. (2016). Curve matching approaches to waveform classification: a case study using icesat data. *GIScience & Remote Sensing*, 53(6):739–758.

[Zhu et al., 2023] Zhu, X., Nie, S., Zhu, Y., Chen, Y., Yang, B., and Li, W. (2023). Evaluation and comparison of icesat-2 and gedi data for terrain and canopy height retrievals in short-stature vegetation. *Remote Sensing*, 15(20).

[Zhu et al., 2022] Zhu, Z., Qiu, S., and Ye, S. (2022). Remote sensing of land change: A multifaceted perspective. *Remote Sensing of Environment*, 282:113266.

UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:
Universidade de Évora
Instituto de Investigação e Formação Avançada — IIFA
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94
7002 - 554 Évora | Portugal
Tel: (+351) 266 706 581
Fax: (+351) 266 744 677
email: iifa@uevora.pt