



Multistage strategy for ground point filtering on large-scale datasets

Diego Teijeiro Paredes¹ · Margarita Amor López¹ · Sandra Buján² · Rico Richter³ · Jürgen Döllner³

Accepted: 30 July 2024
© The Author(s) 2024

Abstract

Ground point filtering on national-level datasets is a challenge due to the presence of multiple types of landscapes. This limitation does not simply affect to individual users, but it is in particular relevant for those national institutions in charge of providing national-level Light Detection and Ranging (LiDAR) point clouds. Each type of landscape is typically better filtered by different filtering algorithms or parameters; therefore, in order to get the best quality classification, the LiDAR point cloud should be divided by the landscape before running the filtering algorithms. Despite the fact that the manual segmentation and identification of the landscapes can be very time intensive, only few studies have addressed this issue. In this work, we present a multistage approach to automate the identification of the type of landscape using several metrics extracted from the LiDAR point cloud, matching the best filtering algorithms in each type of landscape. An additional contribution is presented, a parallel implementation for distributed memory systems, using Apache Spark, that can achieve up to 34× of speedup using 12 compute nodes.

Keywords LiDAR point clouds · Landscape identification · Ground filtering · Apache spark

✉ Diego Teijeiro Paredes
diego.teijeiro@udc.es

¹ Computer Architecture Group, Depto. de Ingeniería de Computadores, CITIC, Universidade da Coruña, Laboratorio 1.2, Facultad de Informática, Campus de Elviña s/n, 15071 A Coruña, Spain

² Departamento de Tecnología Minera, Topografía y de Estructuras, Universidad de León, León, Spain

³ Hasso Plattner Institute, Faculty of Digital Engineering, University of Potsdam, Potsdam, Germany

1 Introduction

At the present time, the transversal role of ALS (Airborne Laser Scanning) point clouds is unquestionable in large-scale geospatial applications such as agroforestry, archeology, 3D urban characterization or landslide recognition [1–4]. The transversality of LiDAR (Light Detection and Ranging) data has been made possible thanks to the rapid development of 3D acquisition techniques and data processing software (specialized tools for extraction, manipulation and analysis of 3D data) [5, 6]. These technological developments together with the multi-purpose character of ALS point clouds have resulted in many geographical national agencies collecting and delivering large size 3D point cloud datasets covering large areas. In Spain, the IGN (*Instituto Geográfico Nacional*) has gone from providing LiDAR data with a nominal point density of 0.5–2 points/m² (second LiDAR cover, 2016–2021) to delivering data with a nominal point density of 5 points/m² (third LiDAR cover, 2022–2025), although, depending upon the region, higher densities may be available (14 points/m² in the region of Navarra).

A major step in the use of LiDAR data is per-point classification to better address the needs of the applications based on it, especially the differentiation of ground and non-ground points. By point classification, we refer to the process of assigning a class to each point from a wider set of possible classes, not limited to ground or non-ground. This differentiation is a critical step in multiple applications since it allows to clearly identify the ground level for further applications [7, 8].

The task of point classification has been around for years, and there has been large amounts of work done in improving the existing proposals [9], either to increase accuracy, defined as the ratio of points correctly classified in several reference or benchmark datasets; or performance, defined as time required to classify a set. Improving the precision of the classifiers increases the quality of the results, which can have benefits in further processes. Increasing the performance allows to finish the classification task faster, or classify larger point clouds in reasonable amounts of time.

As a result of the longevity of this process, there is a plethora of different approaches to point cloud classification in the literature [10]. All of those proposals can be categorized in different ways, according to different features. We can differentiate four types of classifiers: geometric classifiers, traditional machine learning classifiers, modern machine learning classifiers, and hybrid classifiers that use geometric and machine learning together.

Geometric classifiers use purely heuristic algorithmic approaches to perform the classification of the points, using the data contained in the point cloud or derived from it. Multiple analyses can be performed on the geometric data in the point cloud, such as local point density, slope, elevation, normal vectors or coplanarity [11]. A critical step in most of these classifiers is segmentation, a process in which the point cloud is divided into multiple homogeneous regions, and there are multiple techniques for it, from edge detection [12] to region growing [13], model fitting techniques and unsupervised clustering [14].

The second type uses machine learning techniques from other fields, like image processing. The typical examples are creating images and then applying image segmentation techniques followed by assigning the class from each pixel to all of the points that the pixel is representing. There are multiple examples of these classifiers, using several approaches like support vector machines (SVM) [15], Markov Random Fields (MRF) [16] and Conditional Random Fields (CRF) [17]. There are also some established approaches, AdaBoost-based proposals are common [18], and CapsNet deserves a special mention since it is the basis of several other approaches [19].

The third type of classifiers fully exploit the recent advancement in machine learning using new techniques and exploiting the point data directly, instead of converting it to other types of data, a task that was very challenging due to the size of the data. However, the surge of deep learning opened the way to exploit this path, with competitive results compared to other proposals. One of the most important classifiers of this type is PointNet [20] and its derivatives such as PointNet++ [21], due to being one of the first to use point data directly in a neural network, and being the base for multiple improvements and extensions, becoming a sort of reference when benchmarking new proposals. Multiple proposals in this category exploit the convolution operation, some generating multiple images from different viewpoints to apply a more traditional convolution layer, for example, SnapNet [22], or using a voxelized version of the data like VV-NET [23]. Some works focus their efforts on the application of machine learning for geospatial point clouds to build geospatial digital twins at several scales, from indoor models to virtual 3D city models [24].

The last category encompasses approaches that combine machine learning with geometric approaches, and as such it can cover vastly different proposals, but there are fewer examples. Some proposals are more similar to geometric approaches, for example, using traditional region growing to segment the point cloud, then using a SVM to classify each region [25]. Other proposals use more complex steps, for example, combining Morphological Profiles and Convolutional Neural Networks [26].

Ground point filtering, the differentiation of ground and non-ground points, has specific characteristics that can be exploited to improve the results. As a result, ground point filtering has dedicated research as well as algorithms, usually in the geometric category. In this field, hybrid methods usually refer to those methods that combine other ones in some way, with no regard to the type of each method combined. In the literature, there are multiple classifiers based on morphological filters, such as the Simple Morphological Filter (SMRF) [27] used by the Point Data Abstraction Library (PDAL), the displacement segmentation filter [28] or the LIDAR2MDTPlus [29] algorithm. Other approaches are based on the surface method such as the Cloth Simulation Filter (CSF) [30], expectation maximization [31], Progressive Triangulated Irregular Network [32] used by Lastools or the combination of multiple methods [33]. A review of the updated ground filters can be found at [34].

Another challenge in this process is that different classifiers are more suited for some types of landscape and data sources. In [35] an in-depth study of several ground point filtering algorithms is shown on data from airborne LiDAR as well

as UAV (Unmanned Aerial Vehicle) photogrammetry-based point clouds, including some of the ones used in this work, while another study on the impact of different characteristics is shown on the performance of different classifiers in [36]. Most studies into point classification are focused on one type of landscape, with urban landscapes being the most frequent. This responds to the fact that urban landscapes present more types of non-ground objects (buildings, vehicles, low and high vegetation, or road signaling) in higher concentrations in the same area, they are a more challenging landscape with more potential applications of the filtered point cloud.

On a larger scale, county-, province-, state- or even nation-wide, multiple types of landscape are present. These massive point clouds present additional challenges in the ground point filtering task due to the sheer size of the data (excessively long execution times, around 11 days for the complete dataset used in this article) and due to the heterogeneity of the different landscapes. Using only one classifier is not the best for every case. In some landscapes, like mountainous regions with little vegetation, most classifiers offer worse precision in the results compared to other landscapes, and different classifiers have wildly different results.

Our proposal consists of a multistage approach for ground filtering, using the geometric data contained in the point cloud to identify the type of landscape present in each area. The classifier and configuration to use for each area is selected according to the type of landscape identified in the area.

The sizes that modern datasets can reach are large enough, with 5 Terabytes (TB) of compressed data and over 130 billion points for the complete dataset used in this work, to exceed the computational capabilities of current computers. Some distributed computing proposals have been presented to exploit the computational power of distributed memory systems for point classification and other tasks [37–39]. Our proposal can make use of these distributed memory systems to increase performance, being able to parallelize the execution of the same stage for every area in the dataset. This is achieved through the use of the Apache Spark framework running on a local cluster.

This article details our proposal followed by an analysis of the results obtained. Section 2 introduces the dataset used and the specific areas of study. Section 3 describes in detail each of the stages of the proposal. Section 4 details the distributed memory implementation using Spark. Section 5 shows the analysis of the results obtained from different points of view and Sect. 6 closes the paper with some conclusions and future work.

2 Data sets

Considering that this research is conducted at national-level, which seeks to maximize the use of the compute resources, the PNOA LiDAR data over the region of Navarra [40] is used as benchmark and validation sets of the proposed methods (National Plan for Aerial Orography, hereafter referred to as PNOA) (see Fig. 1). One of the most distinctive characteristics of these point clouds is the point density, 14 points/m², while the point density is less than 4 points/m² in the remaining regions of Spain.

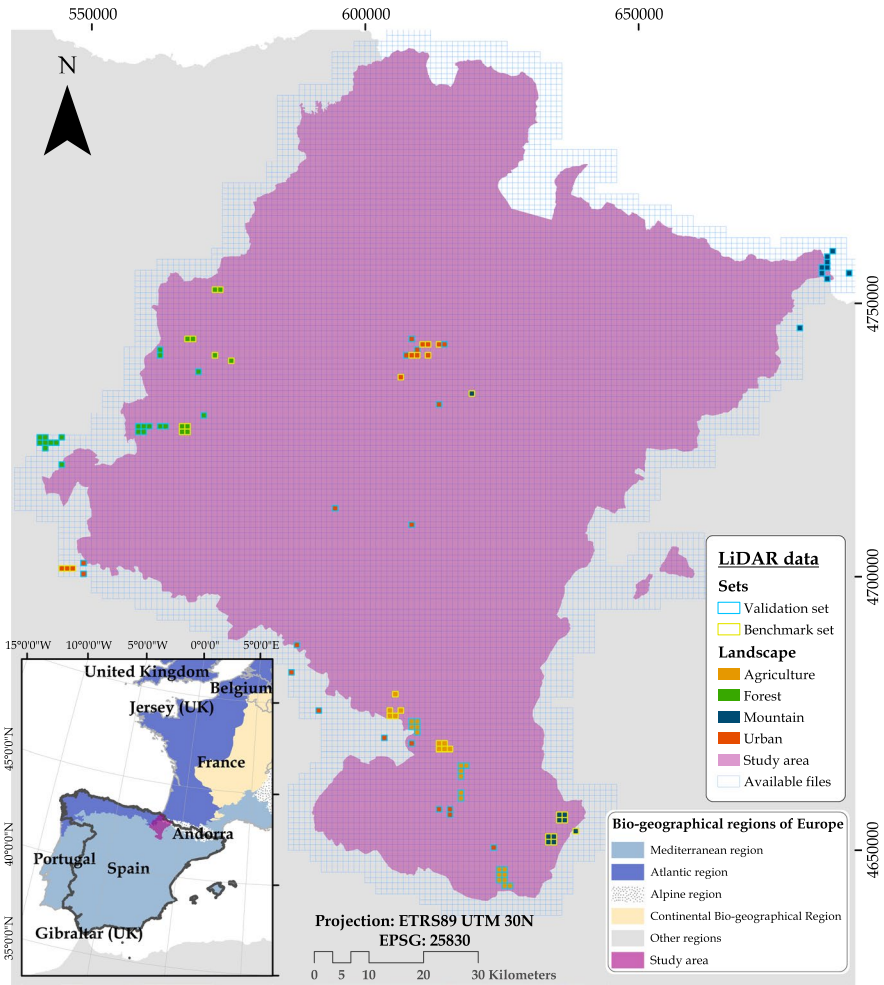


Fig. 1 Study area and the location of each LiDAR data file

LiDAR data used in this study were collected using a Single Photon LiDAR sensor (SPL100) mounted on the Beechcraft B200 King Air aircraft between September and November 2017 and covers the region of Navarra (Spain). The point clouds correspond to the second round of nationwide ALS measurements, which are publicly available in Spain through PNOA. Square LiDAR blocks (files) of 1 km side, with a nominal point density of 14 points/m² were obtained from [40]. Each file (LASzip files, compressed version of LAS files, version 1.4 [41]) contains the points located in the 1 km² area it covers. We will refer to these files as tiles.

The point cloud in each tile includes, in addition to the usual point coordinates, RGB (Red, Green and Blue color channel data) and NIR (Near-InfraRed) attributes for each point, derived from a fast orthoimage collected using an RCD30 camera on

a joint flight with the LiDAR sensor. Although this orthoimage may be displaced several meters in relation to the ALS point clouds, this limitation will not negatively affect the identification of the landscape. Furthermore, the data are already classified using automatic classification with machine learning processing and a subsequent process adjustment before being made public as a product of the PNOA project.

Finally, the data used in this research are stored in 107 tiles, each one covering a $1 \text{ km} \times 1 \text{ km}$ area, for a total of 75 Gigabytes (GB) of data containing over 2 billion points. Forty tiles, 10 for each type of landscape (agriculture, forest, urban, and mountain), were used during the development of the landscape classification process (benchmark set), while another 67 tiles (validation set) were used to evaluate the process developed with the benchmark set. The type of landscape of each tile has been manually identified. Table 1 includes the characteristics of benchmark and validation data sets, and Fig. 1 shows the location of tiles in the region of Navarra.

3 Overall structure

This section details how the proposed system works, including implementation details. Figure 2 shows the preprocessing phase and the multistage approach workflow. Before the execution of the multistage approach, a preprocessing phase is performed to prepare the LiDAR point cloud with the goal of avoiding problems due to the size of the files. This preprocessing can keep the original LAS files or split them if they exceed a defined size (e.g., 1 million points). This preprocessing is due to several factors. For example, some classifiers may have some limitations on the amount of points that they can process at a time, regardless of whether or not that limitation is intentional. More importantly, the size of the tiles plays an important role since it has a very notable impact in the correct identification of the landscapes. On the one hand, tiles that cover areas too large increase the chances on containing multiple types of landscape in different parts of the tile, which results in less clear-cut metrics for the tile difficulting the identification of the predominant landscape, as well as forcing the use of a potential suboptimal classifier in the less dominant types of landscape. On the other hand, tiles that cover areas too small create challenges for

Table 1 Characteristics of the set of tiles used as benchmark and validation for each type of landscape

Sets	Landscape	Points (million)	Density (points/m ²)	Size (GB)
Benchmark	Agriculture	125	12.5	4.7
	Urban	165	16.5	6.3
	Forest	331	33.1	12.6
	Mountain	128	12.8	4.9
Validation	Agriculture	258	12.9	9.7
	Urban	246	13.6	9.4
	Forest	324	16.2	12.3
	Mountain	427	47.4	16.2

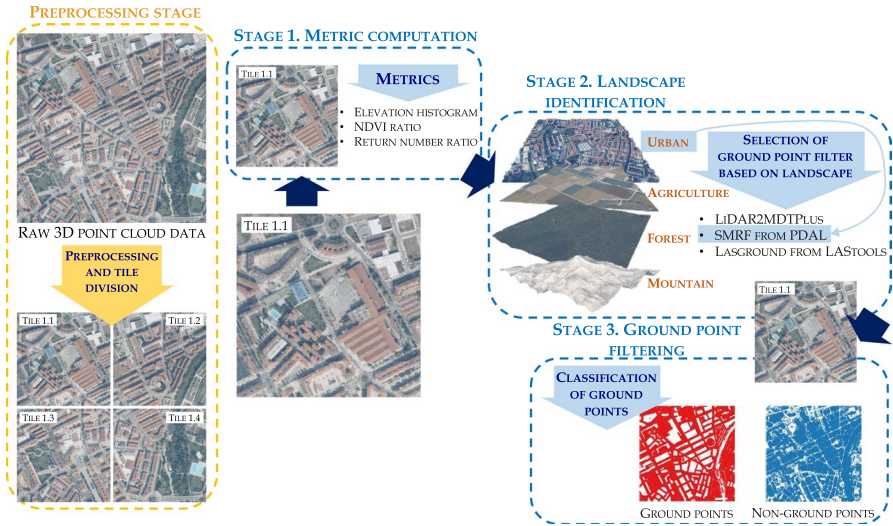


Fig. 2 Preprocessing phase and the multistage approach workflow

landscape identification since they may not contain enough data for the metrics to have a strong basis and can cause misclassifications of the landscape. The best scenario is to have tiles as large as possible containing only one type of landscape. This effect will be explored in Sect. 5.2.

The stages in our proposal are as follows. In the first stage, all of the necessary metrics for the second stage are calculated (see Sect. 3.1). The second stage is landscape identification and algorithm assignment, in which the metrics extracted are used to identify the type of landscape and assign the algorithm or configuration (or several of them) to use for that tile (see Sect. 3.2). In this work, 4 types of landscape are differentiated: agriculture, urban, forest, and mountain areas. In the example tile shown in Fig. 2, the tile is identified as an urban landscape and therefore SMRF done by PDAL is chosen for that tile, represented in the figure with a light blue highlight of the urban landscape and appropriate filter. The third stage is the main stage where the point filtering is performed using the selected algorithm for each tile (see Sect. 3.3).

3.1 Metric computation

Different datasets provide different types of data. Most modern datasets provide color information in the form of RGB data, while a few of them also provide NIR in addition to the RGB data. This variability in data provided is supported in the data format normally used to make the datasets available, the LAS data format. The dataset used in this work uses the LAS data format version 1.4, using the point record format 8 of the standard, which stores both RGB and NIR data for each point, which can be used to get some metrics that cannot be obtained in other datasets.

In this work, three metrics are used: *height histogram*, *NDVI ratio* (normalized differential vegetation index) and *return number ratio*. If NIR data are available, NDVI can be calculated for each point. Alternatively, if NIR field is not available, the intensity field can be used instead. NDVI is a widely used metric for measuring the presence of vegetation [42]

Height histogram is calculated by binning the points by their elevation over the minimum elevation ($\min Z$) of the tile, using 1 m high bins. The height of the bins (1 m) was selected due the speed of computing the corresponding bin for each point and it provides enough differentiation between the types of landscape to be detected. The corresponding bin for a point can be easily calculated by subtracting $\min Z$ from the Z coordinate of the point and truncating the resulting value to the closest lower integer and increasing the count on the corresponding bin. This metric can differentiate between largely flat areas, that show notable peaks in the histogram, and areas with a wide range of elevation, such as areas with steep inclination.

NDVI is an indicator widely used to analyze remote sensing measurement, usually from satellite or aerial photogrammetry, and it can assess whether or not the target contains live green vegetation. With point-wise red and NIR data, the NDVI value can be calculated for each point. The NDVI ratio used in this metric is calculated by obtaining the NDVI for each point using NIR and red channel data according to the following equation:

$$\text{NDVI} = \frac{\text{NIR} - R}{\text{NIR} + R} \quad (1)$$

In the equation NIR and R are the infrared and red color channels, respectively. The range of NDVI is divided into three categories: low ($\text{NDVI} < 0.1$), medium ($0.1 < = \text{NDVI} < = 0.5$) or high ($\text{NDVI} > 0.5$). A low NDVI value is indicative of no vegetation, a medium value indicates some types of soil or growing vegetation, growing crops in the early stages for example, and high NDVI values indicate strong presence of vegetation like tree canopies or fully mature crops. The NDVI is calculated for each point and the counter of points in the corresponding category is increased. The end result is the amount of points in each category of NDVI.

Additionally and specially important for the cases where NIR is not available, *return number ratios* are also calculated. For this metric, we count the amount of points of each return number. Although this data can be extracted from the LAS format header, to avoid using incorrect or outdated data, our proposal computes this metrics by checking each point while computing the other metrics.

3.2 Landscape identification

This stage performs two tasks that are tightly linked. The first task is using the metrics obtained to identify the landscape present in the tile, and the second task is selecting which classifier to use for that landscape.

For the first step, a decision tree based on the categorization of the metrics obtained in the first stage is used. This decision tree is based on knowledge from experts in the field and a visual analysis of the computed metrics, with further tests

of multiple configurations to fine-tune the values of each parameter involved. All of the metrics for each of the benchmark data tiles (see Table 1) are analyzed to develop the conditions to identify each landscape type.

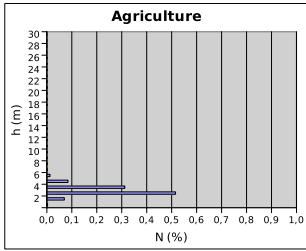
Figure 3 shows the height histograms for one example of each type of landscape, agriculture in Fig. 3a, urban in Fig. 3c, forest in Fig. 3e and mountain in Fig. 3g. The vertical axis displays the bins, and the values are the height over minZ for the tile, in meters. The horizontal axis shows the ratio of points that are in each bin, as a ratio of the total points. If the bar for a bin reaches 0.55, 55% of the points in the tile are located 1 m apart in the vertical coordinate. Please note that not all of the figures have the same horizontal range. In the graphs one bar is drawn for each bin and in high slope tiles there are a lot more bins and they blend together. The goal of the first part of this stage is to detect patterns in the metrics that allow the categorization of them and identify the predominant landscape present in the tile. 10 tiles of each type of landscape are analyzed to identify patterns that differentiate the needed categorizations.

The height histogram for agriculture shows a clear peak of points between 3 and 4 ms above minZ (third bar from the bottom), containing more than 50% of the total points in the tile, and the total vertical range is small compared to the other histograms, all of the points are within 30 ms of minZ. Figure 3b shows an image of one of the agriculture tiles. In the urban tile, the vertical range is larger and the points are more distributed, but there is still a bin with much more points than the rest, at 35 ms above minZ, even though it only contains 13% of the total points. Figure 3d shows an image of one of the urban tiles.

In the forest tile there is no outstanding peak, no bin reaches 2% of the total amount of points, with a more smooth distribution of points, which contrasts with the two previous examples. Figure 3f shows an image of one of the forest tiles. The mountain tile, defined as a high slope area with little vegetation, shows a similar picture to the forest tile, very few bins reach barely over 1%, and there are even more bins increasing the vertical spread. Figure 3h shows an image of one of the mountain tiles.

These height histograms can be categorized into two categories based on the presence of peaks or whether there are large concentrations of points into small ranges of elevation. If peaks are present, the height histogram is classified as a *compact height histogram* and when the points are more uniformly distributed it is classified as *sparse height histogram*. Agriculture and urban areas have compact height histograms since they are flatter areas. Forest and mountain areas have sparse height histograms since they tend to have steeper inclines and in the case of forests on flat land, they will have multiple returns inside the canopy which spread the points along the height of the trees. The exact conditions and values to classify a height histogram as compact or sparse will be studied and tuned in Sect. 5.1.

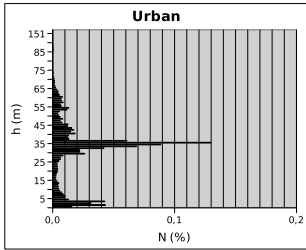
The NDVI metric can be used to detect a relevant presence of vegetation. The top part of Fig. 4 shows the ratio of points in low and medium NDVI ranges, aggregating the data of the 10 studied tiles for each type of landscape. There is not enough points in the high NDVI range to show in the graph. In the agriculture and forest areas the ratio of points in the low NDVI range is noticeably lower, less so in the agriculture areas since the data was captured in autumn where less crops



(a) Agriculture.



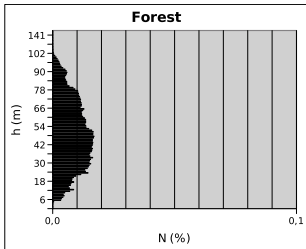
(b) agriculture tile.



(c) Urban.



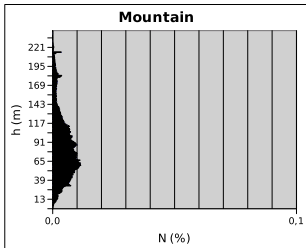
(d) urban tile.



(e) Forest.



(f) forest tile.



(g) Mountain.



(h) mountain tile.

Fig. 3 Example height histograms for a tile of each landscape type

give higher values for NDVI. It is possible to use this to differentiate between tiles with and without vegetation. The tile can be categorized as *vegetation-suggesting* NDVI when the ratio of points in either medium or high is large enough,

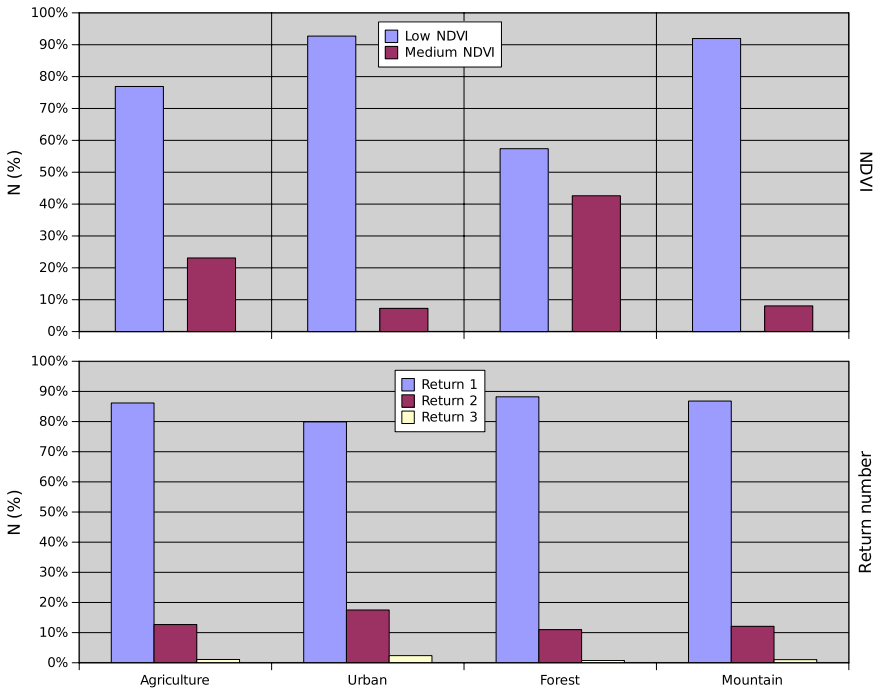


Fig. 4 NDVI and return number ratios for the aggregation of 10 tiles of each landscape

or *bare ground-suggesting* NDVI otherwise, when the vast majority of points fall into the low NDVI category. The exact ratios to classify the tile as *vegetation-suggesting* NDVI will be studied and tuned in Sect. 5.1.

When NDVI is not available, the ratio of points with each return number can be used. The bottom part of Fig. 4 shows the results aggregating the data of the 10 tiles of each type of landscape. The margins are much tighter with this metric, for example mountain areas have more points with higher return numbers than forest areas. This metric is more sensitive than NDVI and therefore we recommend the use of NDVI when it is available.

With all of this data at hand, a decision tree based on the possible categorizations is developed, shown in Fig. 5. This tree has two decisions, the first one uses the height histogram to separate the agriculture and urban tiles from forest and mountain tiles. The second decision uses the NDVI to differentiate between agriculture and urban, and between forest and mountain, depending of the result of the first decision. A more detailed explanation of how this decision tree is used in provided in Sect. 5.1.

Once the type of landscape is identified, the classifier to use for the tile is selected using a direct mapping of landscape type to a classifier. The mapping shown in the figure is the result of the testing performed in Sect. 5.2.

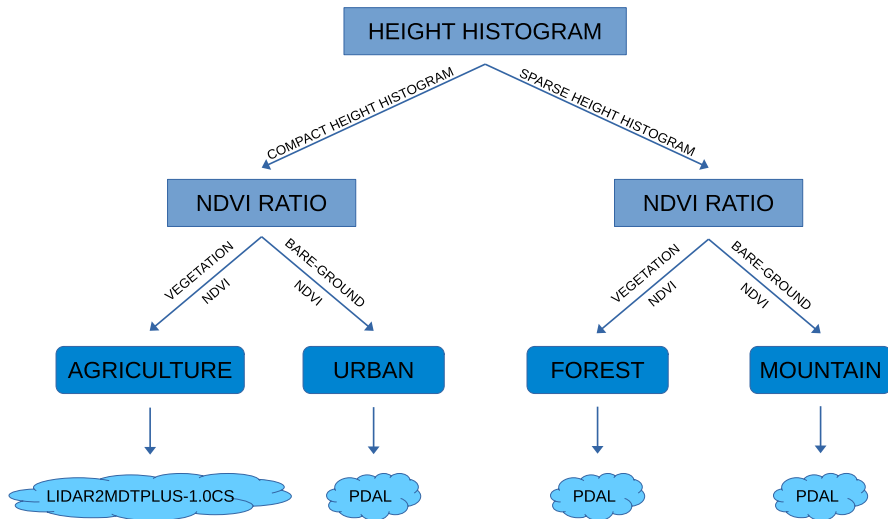


Fig. 5 Decision tree with the classifiers to use in each landscape

3.3 Ground point filtering and assessment

The third stage performs the ground point filtering. The actual inner working of this stage depends heavily on the classifier in use since this stage is tasked with getting the data of the tile ready to be filtered.

In this proposal, three different classifiers are used: LIDAR2MDTPlus [29], SMRF using PDAL [27] and *lasground* from LAStools [43]. These classifiers were selected due to the capability of being able to integrate them into the multistage pipeline and Spark distributed implementation. Both LIDAR2MDTPlus and PDAL are based on morphological filters, exploiting the changes in the elevation of non-ground points after the morphological operation is performed. *Lasground*, on the other hand, uses an adaptive TIN method, using triangulation and densification steps to progressively include all ground points in the triangulated DTM (Digital Terrain Model). Two different configurations for LIDAR2MDTPlus are analyzed, as they behave notably different in each landscape. More algorithms were considered but those were unable to be integrated into the multistage pipeline for several reasons, for example the software ecosystem present in the cluster.

The need to have RGB and NIR values makes it not possible to test on more common datasets such as ISPRS, as it does not have the required data to calculate NDVI for each point.

We have access to LIDAR2MDTPlus' source code, and it is already written in Java so it can be directly integrated into our proposal, loading the data into memory before performing the filtering. PDAL and LAStools are external tools, integrated into our proposal by invoking the command line tools from the Java Virtual Machine. The results with LAStools are obtained by dividing each tile in order to not violate the licensing restrictions.

For the analysis of the the effectiveness a comparison of the results of each classifier is performed, comparing against the point class with the classification for each point provided by the PNOA project, checking that each point has the correct classification, identifying true positives (TP) or correctly classified ground points, true negatives (TN) or correctly classified non-ground points, false positives (FP) or non-ground points incorrectly classified as ground, false negatives (FN) or ground points incorrectly classified as non-ground, and total point count (TPC). With those statistics, the three often used errors are calculated: type I error (omission error, ground points wrongly classified as non-ground $\frac{FN}{FN+TP}$), type II error (commission error, non-ground points wrongly classified as ground $\frac{FP}{FP+TN}$) and total error (wrongly classified points $\frac{FN+FP}{TPC}$) [44].

Each application and use case of the filtered point cloud is more dependent on low error rates of different types and therefore all of the error rates are shown so the reader can make an informed decision based on their particular case. Table 2 shows the error rates for the agriculture tiles used in the previous section. Table 3 shows the error rates on urban tiles. Table 4 shows the error rates on forest tiles, and Table 5 shows the error rates on mountain tiles. The best classifier in each case is indicated by a italics.

It is clear that some classifiers show better results in different types of landscape, with agriculture being a bit more challenging for all classifiers and mountain being a very challenging landscape for all of them. Every classifier has one of the partial

Table 2 Classifier error rates on agriculture tiles

Classifier	Total %	Type I %	Type II %
PDAL	8.17	0.33	25.07
LAStools	9.46	8.44	11.63
LIDAR2MDTPlus-0.25CS	8.99	0.24	27.84
<i>LIDAR2MDTPlus-1.0CS</i>	6.72	4.33	11.89

Table 3 Classifier error rates on urban tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	3.71	2.89	4.50
LAStools	5.69	8.47	3
LIDAR2MDTPlus-0.25CS	6.00	2.31	9.59
LIDAR2MDTPlus-1.0CS	4.79	7.28	2.38

Table 4 Classifier error rates on forest tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	2.16	0.96	2.33
LAStools	5.21	40.41	0.38
LIDAR2MDTPlus-0.25CS	2.44	1.47	2.57
LIDAR2MDTPlus-1.0CS	2.57	15.15	0.85

Table 5 Classifier error rates on mountain tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	12.86	2.08	34.59
LAStools	25.43	34.80	6.55
LIDAR2MDTPlus-0.25CS	12.87	4.49	29.74
LIDAR2MDTPlus-1.0CS	29.09	39.68	7.78

errors much higher than the other, but not all the classifiers agree on which type of error is low. With this information the decision tree in Fig. 5 is completed to include which classifier should be used in each landscape type.

Some external tools provide the results in a LAS format different from input, which causes loss of data in some cases. One example is input in LAS version 1.4 but output is in LAS version 1.2 which does not support NIR data. Some post-processing of the results of the classifiers can be performed to avoid this loss of data, by loading the point data from the input tile, identifying the same point in the output of the classifier and overwriting the classification on the input tile data with the class of the point taken from the output of the classifier.

The process of identifying the same point in both files is performed using the point coordinates, allowing a configurable margin of error, set to 1 cm for the dataset used in the test with a point density of 14 points/m². Nonetheless, not all of the points are matched sometimes and some points are lost in the classification process, but this amount is small enough to not be too relevant. In order to keep that amount as low as possible, after all of the points have been tested for a match, a final match is attempted on every remaining point increasing the margin to 10 cm.

4 Distributed computing execution

While all of the processes described to this point can be executed using traditional data structures from Java like lists, the pipeline is designed to be executed in parallel through Apache Spark. Apache Spark is a parallel framework that allows the use of distributed memory system to parallelize the execution of pieces of code on large amounts of data. Spark achieves most of its goals through the use of Resilient Distributed Datasets or RDDs, an abstraction similar to a list that represents a set of elements distributed among the compute nodes available. Operations executed on the elements of this RDD are executed in parallel in the compute nodes that hold the elements, each node performing the operation on the elements it is storing. As long as there is enough memory to store the data in the RDD aggregating the memory available on all of the nodes, the data is kept in memory accelerating the execution of the operation, persisting and recovering from disk data when the total size is too large, providing a transparent out-of-core computation on very large datasets.

RDDs are also the basis to other benefits that Spark provides. All of the operations executed on the RDD are added to a graph and when a task fails, Spark automatically recovers from the error by reconstructing the part that was lost using this

graph, executing the necessary operations since the last persisted state of the lost part of the RDD. This makes Spark an excellent choice for parallelizing the processing of large amounts of data, as it provides fault tolerance with no extra effort.

Our multistage approach is designed to leverage Spark to parallelize each stage. This is performed by creating an RDD with the paths of every input tile and performing each stage as a map operation on the RDD. Figure 6 shows how the RDD evolves during the entire pipeline. Each map and reduce operation (the transitions between RDDs) can be applied at the same time on all elements of the RDD as long as there are enough resources available.

This parallel execution puts a lot of strain on the filesystem due to all of the reading operations required to perform each of the stages, specially in the filtering stage with external tools where writing is also involved. The input and output folder are required to be accessible by all computer nodes, using a network accessible filesystem. This is required since the input tiles are read in several stages by the worker nodes, and also written to disk in the filtering stage to be able to use external tools.

Spark follows the design of moving the computations to the location of the data. The distribution of the data in the RDDs is what allows this feature, however, it requires the data to be completely loaded into a RDD. In our proposal, we need to use external tools that cannot access the data in the RDDs from outside of the processes managed by Spark, so those external tools are not aware of the locality of the data. It could still be exploited by exporting the data from the RDD to local disk before using external tools, introducing the I/O overhead once again and negating most of the benefits.

Some measures were taken to reduce the I/O overhead. The proposal uses local storage on the compute node for storing input and output data for external tools, avoiding the I/O traffic caused by the filtering process through the network. For each task that needs to use external tools, the input tile is written on local storage in the

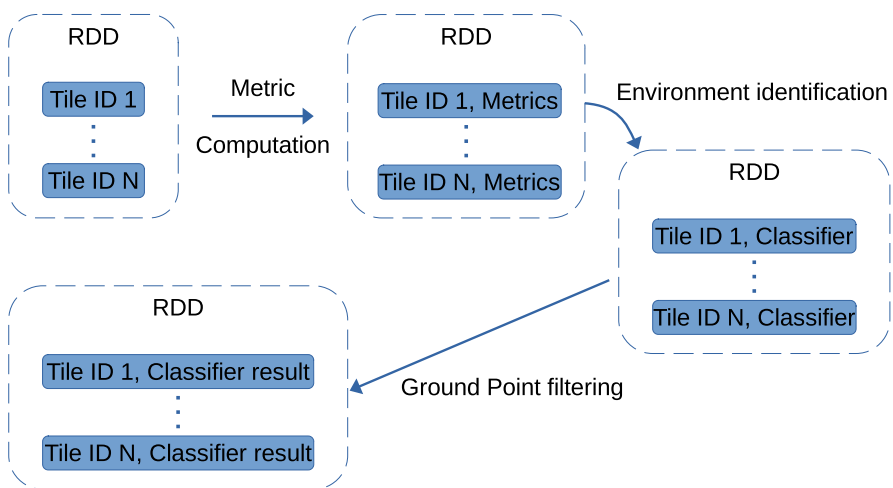


Fig. 6 Evolution of the RDD during the multistage approach execution

node that will execute the external tools, and the final result is moved, writing it in the shared filesystem that stores all of the filtered tiles resulting of the filtering stage. Replacing the external tools with other classifiers more tightly integrated in the pipeline helps to reduce the I/O overhead since less data needs to be written to disk to be filtered, it can be done directly in memory.

All of the processing in the different stages other than the execution of the classifiers themselves is performed with an out-of-core strategy, reading and writing to disk in chunks to keep memory consumption low. The size of the chunks is defined as a set amount of points read/written each time and the exact amount is adjusted to allow the concurrent processing of multiple tiles without exceeding available memory when Spark executes multiple tiles in the same node.

5 Performance analysis and results

In this section, the performance of our multistage approach is analyzed from different points of view. In Sect. 5.1, the first two stages are evaluated with an analysis of the landscape identification success rate. In Sect. 5.2, the complete multistage approach is assessed in accordance with the accuracy of the ground filtering process. To close the section, the Spark implementation is evaluated in Sect. 5.3 by measuring the time required to perform the ground filtering on all of the tiles used in this work using different amounts of resources.

Throughout this section and in view of the characteristics of LiDAR data (see section 2), we can compare the results of our multistage approach to the classification in the original data to measure the errors in classification. This dataset includes overlap points, sensor noise and other problematic data, but it is clearly labeled using custom LAS classes for the points and is simple to remove those noisy points simply removing the points with certain classes in the preprocessing phase.

5.1 Landscape identification

The first set of tests are focused on the two first stages of our proposal. The goal is to evaluate the capabilities of the metrics and decision tree used. The testing is performed on 10 tiles of each type of landscape as benchmark, manually selected and classified using satellite image. Table 1 shows the characteristics of the set of tiles used for each type of landscape, number of points and memory requirements. As indicated above, each tile occupies an area of 1 km², so 40 km² are being processed. It should be noted that, in general, the tiles are located in different zones of Navarra (see Fig. 1). These tests have a double purpose, fine tuning the values of the parameters and conditions used in the decision tree and show the capabilities of the second stage in detecting the type of landscape present, both with and without optimized values for the parameters.

On each step of the decision tree, there are several conditions according to the metrics presented in Sect. 3.1 to classify the metric toward one type of landscape. Different values for each parameters of those conditions are tested and the success

rate of the second stage is evaluated, the one that performs the type of landscape detection. More than 60 different configurations were tested, only the most representative are shown in this work. The descriptions of the conditions used in each step of the decision tree as well as the terminology used to reference each parameter are detailed in the next paragraphs while the values used in each configuration tested are shown in Table 6.

For the first step of the decision tree, the one that categorizes the height histogram, a couple of definitions should be provided first. The histogram *peak* is defined as the bin with the largest amount of points. A *close to the peak* bin is defined as a bin that contains more than a set percentage of the points of the *peak* bin, said percentage being named *histogram peak percentage (hpp)*. A *minor bin* is defined as a bin with less points than a set percentage of the total, named *minor bin limit (mbl)*. With those terms defined, the conditions checked to categorize the height histogram are as follow: A histogram is directly categorized as *compact* if there are less than 30 bins, once those bins with less than 5 points are discarded to eliminate outliers that disrupt the height histogram. A histogram is directly categorized as *sparse* if the *peak* contains less than a set percentage of the total points, referred as *peak minimum percentage (pmp)*. A histogram is directly categorized as *sparse* if the *minor bins* accumulate more than a set percentage of the total points, referred as *minor bins aggregated ratio (mbar)*. Lastly, a histogram is categorized as *compact* if the percentage of bins *close to the peak* exceeds a set amount named *percentage of peak bins (ppb)* and it is categorized as *sparse* otherwise.

For the second step in the decision tree, the one that uses the NDVI or return number metrics, one conditions for each metric is applied. If the NDVI is used, a tile is categorized as *vegetation-suggesting NDVI* if the sum of points with medium or high NDVI values exceeds a set percentage of the total points, named *ndvi_veg*. When the return number is used, a tile is classified as *vegetation* if the points of second return exceeds a set percentage of the total, named *nr2*, or the the points of third return exceeds different set percentage of the total points, named *nr3*.

An explanation of the classification of each tile is as follows: the height histogram and NDVI ratios for the tile are calculated. The tile is then categorized as *compact height histogram* or *sparse height histogram* using the corresponding values of the parameters: *hpp*, *ppb*, *pmp*, *mbar* and *mbl*. The NDVI ratios are used to categorize the tile as *vegetation-suggesting NDVI* or not using the *ndvi_veg* parameter if the NDVI data is present, otherwise the return number ratios are checked using the *nr2* and *nr3* parameters. The tile is then classified following the decision tree of Fig. 5,

Table 6 Values, in percentages, used for the categorization of the metrics in the decision tree

ID	<i>hpp</i>	<i>ppb</i>	<i>pmp</i>	<i>mbar</i>	<i>mbl</i>	<i>ndvi_veg</i>	<i>nr2</i>	<i>nr3</i>
1	75	10	1	80	1	15	–	–
2	75	10	1	80	1	–	20	10
3	75	10	1	80	1	10	–	–
4	75	5	1	80	1	15	–	–
5	–	–	5	–	–	13	–	–

the type of height histogram is checked, if it is a *compact* histogram, it will be either an agriculture or urban tile. If it has a *vegetation-suggesting* NDVI it is categorized as agriculture, otherwise it is categorized as an urban tile. If the height histogram was *sparse*, it can be either a forest or mountain tile. If it has a *vegetation-suggesting* NDVI it is categorized as forest, otherwise it is categorized as a mountain tile.

Table 6 shows the exact values used for these parameters in each configuration. Table 7 shows the results in landscape identification with a confusion matrix using overall accuracy as the metric to base the decision of which configuration is better. The table shows the amount of tiles of each type of landscape detected for each set of tiles, for the configurations tested. For each landscape type shown in the first row of Table 7, it displays the amount of tiles classified as agriculture (A), urban (U), forest (F) and mountain (M). The first column of both tables contains an identification number for referencing the configuration used between the tables and the following list describing them in text. The last three columns show the correct identification rate on the two steps of the decision tree (S1 indicates the success rate of correctly categorizing the height histogram as *compact* or *sparse* and S2 indicates the success rate of correctly categorizing the NDVI ratios or return number ratios as *vegetation-suggesting*), as well as the final identification rate on the last column (S3).

Of all of the combinations used, the one that shows the best results is the fifth one, which checks whether or not any bin of the tile exceeds 5% of the total points. This is a condition that differentiates the flat tiles from the inclined tiles with an accuracy of 100% in the 40 tiles tested.

For the second step of the decision tree, NDVI shows better results, with a correct identification on the second step of up to 92.5%. Agriculture and mountain regions are more challenging. The only misclassified agriculture tile is caused by the presence of crops with lower NDVI values in the time of year at the moment of data acquisition. Figure 7 shows the tile in question using the RGB data in the point cloud. While it is evident that it is an agriculture area, the vegetation present does not reach the threshold for medium NDVI values, the fields around the center of the image being a good example. In the case of the misclassified mountain tiles, it is due to the presence of enough vegetation to reach the NDVI threshold, even though the

Table 7 Landscape identification for each landscape type

ID	Agriculture				Urban				Forest				Mountain				Total (%)		
	A	U	F	M	A	U	F	M	A	U	F	M	A	U	F	M	S1	S2	S3
1	8	2	0	0	0	10	0	0	0	0	10	0	0	8	1	1	80	92.5	72.5
2	6	4	0	0	3	7	0	0	0	0	2	8	6	2	0	2	80	47.5	42.5
3	9	1	0	0	2	8	0	0	0	0	10	0	1	7	2	0	80	85	67.5
4	8	2	0	0	0	7	0	3	0	0	10	0	0	8	1	1	72.5	92.5	65
5	9	1	0	0	0	10	0	0	0	0	10	0	0	0	2	8	100	92.5	92.5

For each metric configuration (ID rows), the number of landscape types is indicated: Agriculture (A), Urban (U), Forest (F) and Mountain (M). The total percentage of succes on the first stage of the tree (S1), the second stage of the tree (S2) and the whole tree (S3) is indicated in the last columns



Fig. 7 Misclassified agriculture tile, classified as urban, shown using the RGB data of each point

slope is high enough to be classified as mountain. These two cases, agriculture tiles with non green types of crops or mountain with forest present, would be subclassifications of the landscapes used in this paper. However, we have left those tiles in our analysis in order to test our proposal in some borderline cases. Figure 8 shows one of the tiles using the RGB data in the point cloud. The configuration in row 5 of Table 7 is the one that will be used for the rest of the tests.

5.2 Classification results

In this section, the accuracy of the whole multistage approach is analyzed on the final results, ground filtering. All of the stages are performed in each set of tiles, extracting the metrics, identifying the landscape and running the best classifier for the landscape detected, for each tile. The results are measured by comparing the classification produced by our multistage approach with the original classification provided by the dataset provider.



Fig. 8 One of the misclassified mountain tiles, classified as forest, shown using the RGB data of each point

Table 8 Error rates of the different classifiers and the multistage approach on agriculture landscape using the benchmark tiles

Classifier	Total %	Type I %	Type II %
PDAL	8.17	0.33	25.07
LIDAR2MDTPlus-0.25CS	8.99	0.24	27.84
<i>LIDAR2MDTPlus-1.0CS</i>	6.72	4.33	11.89
Multistage	6.84	4.27	12.37

Table 9 Error rates of the different classifiers and the multistage approach on urban landscape using the benchmark tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	3.71	2.89	4.50
LIDAR2MDTPlus-0.25CS	6.00	2.31	9.59
LIDAR2MDTPlus-1.0CS	4.79	7.28	2.38
<i>Multistage</i>	3.71	2.89	4.49

Table 10 Error rates of the different classifiers and the multistage approach on forest landscape using the benchmark tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	2.16	0.93	2.33
LIDAR2MDTPlus-0.25CS	2.44	1.47	2.57
LIDAR2MDTPlus-1.0CS	2.57	15.15	0.85
<i>Multistage</i>	2.16	0.93	2.33

Table 11 Error rates of the different classifiers and the multistage approach on mountain landscape using the benchmark tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	12.86	2.08	34.59
LIDAR2MDTPlus-0.25CS	12.87	4.49	29.74
LIDAR2MDTPlus-1.0CS	29.09	39.68	7.78
<i>Multistage</i>	12.86	2.08	34.59

Type I and Type II reference errors are calculated, as well as the total error. Applications that use ground filtering as a step preferably use a type of filter that minimizes one of the above errors, even if the other error is increased. For example, in power line detection applications, ground points are removed and only non-ground points are processed, so low Type II errors are critical to avoid removing points that need to be processed. However, in applications for railway infrastructure, low Type I errors are desired, as the relevant points are the ground points to analyze the condition of the terrain along the track.

Tables 8, 9, 10, 11 and 12 show the tests performed on the same set of tiles used in the previous section, called *benchmark tiles*, with 10 different tiles of each type of landscape. LAsTools is not considered in these tests since it was not found the best in any of the landscapes. The best classifier is highlighted by a

Table 12 Error rates of the different classifiers and the multistage approach aggregating all four landscapes using the benchmark tiles

Classifier	Total %	Type I %	Type II %
PDAL	5.33	1.64	7.69
LIDAR2MDTPlus-0.25CS	6.10	2.23	8.57
LIDAR2MDTPlus-1.0CS	8.28	16.98	2.73
<i>Multistage</i>	5.11	2.79	6.59

italics in its row based on the total error, highlighting the best total error for each case in bold text.

These tables show that our multistage approach obtains the same result as the best classifier for each case, except for agriculture regions. In agriculture tiles, the misclassified tile causes the multistage approach to use PDAL in that tile, which results in a difference in the errors in the direction of the values achieved by PDAL.

The results for mountain tiles show a high error rates in some of the errors types checked, and therefore it is more difficult to name one classifier as best. Due to the fact that the misclassified mountain tiles are identified as forest, which also uses the same filtering algorithm (PDAL), our multistage approach achieves the same error rates as PDAL. When all 40 tiles are aggregated, the results for the multistage approach are better than using only one classifier in all of the tiles, this is the main goal of the proposal.

To evaluate the selected metrics and conditions, a set of validation tiles are incorporated to each type of landscape (see Table 1), tiles that were not used during the development of the metrics. Table 13 shows the type of landscape detected, both for benchmark and validation tiles, while Tables 14, 15, 16, 17, and 18 show the ratio of errors when these tiles are incorporated. The best configuration from Table 7 is able to detect the correct type of landscape in an automated way with a success rate of 91% on these additional tiles, with an overall success rate of 91.5% when both benchmark and validation tiles are considered.

The results when the validation tiles are incorporated remain similar to the results with only benchmark tiles, although the results for forest tiles deserve further explanation. With the additional tiles, the best classifier according to the total error for the

Table 13 Landscape identification success rate

	Tile set	Landscape detected					Success %
		AG	UR	FO	MO	Total	
Benchmark	Agriculture	9	1	0	0	10	90
	Urban	0	10	0	0	10	100
	Forest	0	0	10	0	10	100
	Mountain	0	0	2	8	10	80
Validation	Agriculture	20	0	0	0	20	100
	Urban	2	13	1	2	18	72.22
	Forest	1	0	19	0	20	95
	Mountain	0	0	0	9	9	100

Table 14 Error rates of the different classifiers and the multistage approach on agriculture landscape using the benchmark and validation tiles

Classifier	Total %	Type I %	Type II %
PDAL	7.43	0.32	27.11
LIDAR2MDTPlus-0.25CS	8.62	0.28	31.71
<i>LIDAR2MDTPlus-1.0CS</i>	6.66	4.39	12.96
Multistage	6.7	4.38	13.15

Table 15 Error rates of the different classifiers and the multistage approach on urban landscape using the benchmark and validation tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	5.66	3.42	8.30
LIDAR2MDTPlus-0.25CS	8.08	2.39	14.79
LIDAR2MDTPlus-1.0CS	6.99	9.79	3.67
Multistage	5.93	4.02	8.18

Table 16 Error rates of the different classifiers and the multistage approach on forest landscape using the benchmark and validation tiles

Classifier	Total %	Type I %	Type II %
PDAL	3.95	1.17	4.40
LIDAR2MDTPlus-0.25CS	4.49	1.45	4.98
<i>LIDAR2MDTPlus-1.0CS</i>	3.19	12.54	1.70
Multistage	3.94	1.26	4.37

Table 17 Error rates of the different classifiers and the multistage approach on mountain landscape using the benchmark and validation tiles

Classifier	Total %	Type I %	Type II %
<i>PDAL</i>	23.70	18.83	38.99
LIDAR2MDTPlus-0.25CS	26.84	25.23	31.88
LIDAR2MDTPlus-1.0CS	59.24	76.65	4.55
<i>Multistage</i>	23.70	18.83	38.99

Table 18 Error rates of the different classifiers and the multistage approach aggregating all four landscapes using the benchmark and validation tiles

Classifier	Total %	Type I %	Type II %
PDAL	10.43	8.75	12.16
LIDAR2MDTPlus-0.25CS	12.20	11.19	13.24
LIDAR2MDTPlus-1.0CS	20.15	36.25	3.62
<i>Multistage</i>	10.35	10.02	10.69

whole set of tiles is *LIDAR2MDTPlus-1.0CS*, while the best one was PDAL when only the benchmark tiles were considered. However, even if *LIDAR2MDTPlus-1.0CS* achieves a lower total error, Type I error is notably higher than PDAL's Type II error, exceeding 12%, while PDAL achieves both partial errors below 5%. For this

reason, we consider PDAL a better balanced classifier, and preferred for a general use of the filtered point cloud. Since the one used by the multistage approach on forest tiles is still PDAL, the results achieved by the multistage approach are very similar to the ones achieved by PDAL, since it is the classifier selected for all but one tile, as shown in Table 13. 10 benchmark tiles and 19 out of 20 validation tiles are classified as forest and use PDAL, while the remaining tile is classified as agriculture and *LIDAR2MDTPlus-1.0CS* is used in that tile, causing the difference in Type I and 2 errors, in the direction of the values achieved by *LIDAR2MDTPlus-1.0CS*. Both PDAL and Multistage rows are highlighted in a lighter gray to indicated that, while not the best result in this particular set of tiles, PDAL is the reference for forest tiles.

Once again, our multistage approach is capable of achieving the best total error when all of the tiles are aggregated. The proposed system is capable of matching the designated best classifier in each type of landscape using an automated process, with a maximum deviation on the total error of 0.27%.

5.3 Distributed computing performance

To close this section, the performance that Spark achieves is analyzed. For this analysis, we use a cluster with 13 nodes, each one equipped with two Intel Xeon E5-2660 Sandy Bridge-EP with 8 cores each at 3.0 GHz for a total of 16 cores, 64 GB of memory and one 1 TB local hard disk drive. Multiple sets of tiles are used to get a more comprehensive analysis. The files are hosted in a shared filesystem accessible by all of the compute nodes.

Table 19 shows the execution time of the complete multistage pipeline in hours (T), the speedup (SP) achieved of the parallelization for different amounts of computer nodes, reserving one node for the Spark driver to reduce the interference with the nodes performing the classification, and the performance in millions of points processed per hour (pt/hr). Each configuration is named with the number of nodes used (N) and the total number of cores (C) available for performing the filtering. The sequential configuration performs the processing without using Spark, using only one processing thread that performs each stage for all of the tiles before progressing

Table 19 Spark implementation execution times and speedups on Plutón. T shows the time in hours and SP shows the speedup compared to sequential

	Configuration	B			B+V			4-split B		
		T	SP	pt/hr	T	SP	pt/hr	T	SP	pt/hr
	Sequential	8.20	–	90	18.39	–	109	7.45	–	101
Spark	1N 16C	2.24	3.67	335	2.97	6.18	673	1.07	6.98	703
	2N 32C	1.79	4.6	420	2.26	8.15	887	0.54	13.79	1387
	4N 64C	0.96	8.53	779	1.89	9.75	1062	0.37	20.22	2034
	8N 128C	0.93	8.8	803	0.92	19.96	2173	0.26	29.11	2927
	12N 196C	–	–	–	–	–	–	0.22	34.32	3452

to the next stage. B test uses the Benchmark tiles, 40 tiles described in Table 1. B+V test uses as input data the Benchmark+Validation tiles, also using tiles described in Table 1, for a total of 98 tiles. 4-split B test shows the results when each tile of the benchmark set is divided into 4 by dividing each axis in half, totaling 160 tiles. With sufficient computational resources to perform the parallel classification of each tile, the total execution time of a test is the processing time of the slowest tile. To explore the impact of this time on parallelization performance, each tile was split into 4 smaller tiles to reduce the number of points in each tile with the goal of reducing the execution time of the slowest tiles.

The speedup achieved in the B test stagnates around 8.5x using 4 nodes, with little improvement going to 8 nodes since with only 40 tiles, 4 nodes already have enough cores to dedicate one to each tile. The speedup is therefore dictated by the time required to complete the slowest tile.

To analyze these results, Fig. 9 shows the time required to process each of the tiles in each set. These tiles are not balanced in terms of the run time required to filter the ground points. With few tiles, the slowest tile quickly becomes the bottleneck, while in cases with a larger number of tiles, there is a possibility that the slowest tiles are grouped with the fastest, spreading the heavier workloads among different nodes.

When the size of the tiles is reduced, the execution time required to process each one is also reduced. In the 4-split B Test, the capabilities of our proposal is proved reaching speedups of 34x. This is due to there being no tiles that dominate the execution time too much. If a large enough number of tiles is processed, no individual tile represents a large part of the total processing time, as well as allowing more balanced workloads.

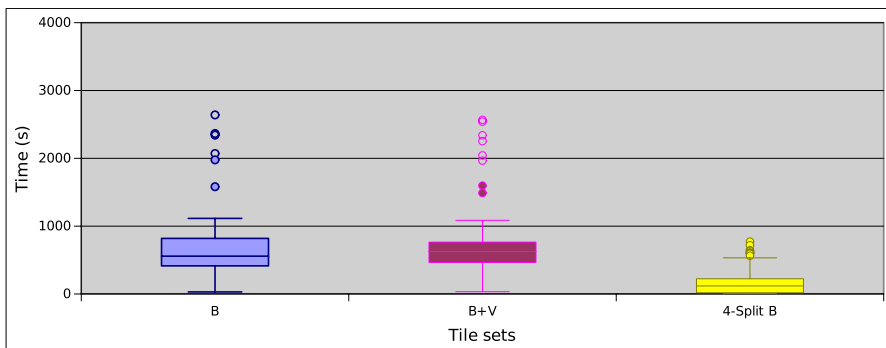


Fig. 9 Times required to filter each tile in each set

6 Conclusions and future work

A multistage approach for ground point filtering for automatic territory-level classification with no human intervention is presented in this paper, requiring low execution times thanks to the parallelization using Spark. Our proposal uses several stages before performing the ground point filtering to identify the type of landscape for each tile and uses a classifier well suited for each tile. The first stage extract several metrics from the point cloud data, metrics that are used in a second stage to identify the type of landscape present. A third stage performs the ground point filtering using the configured classifier for the type of landscape found. A Spark implementation is also presented, allowing the use of distributed memory systems to reduce the time required to perform the ground point filtering. This is thanks to the implementation of each stage as a map operation in Spark, using RDDs that transfer the relevant data between the different stages. Spark performs the execution of each stage in parallel in multiple tiles.

The results achieved in the landscape detection stage show a 92.5% of success rate on the benchmark tiles, and an overall 91.5% correct landscape identification when additional unseen tiles are incorporated to the dataset. The results achieved in the ground point filtering when using the multistage approach are similar with the best classifiers in each type of landscape. While, due to its nature, the presented multistage approach cannot improve the results of the best classifier when looking at only one type of landscape, it allows the automatic use of the best classifier for the landscape present on large-scale datasets where multiple types of landscape are present, improving the results of the entire dataset.

The Spark implementation presented allows a speedup of the entire multistage approach up to 34 times in a distributed memory system using 12 compute nodes. The performance ceiling is highly dependent on processing time required by the slowest tile in the set.

The work ahead begins with exploring more metrics and classifiers, in order to increase the diversity of landscapes that can be detected, as well as increasing the accuracy of the landscape identification, critical to achieve the best classification results. There are some that will be studied such as horizontal coplanarity, normal vector diversity, local point density, slope and elevation, but not limited to those. With regards to classifiers, our focus will be incorporating different classifiers that use different algorithms, as well as looking for more specialized classifiers as long as the landscape they specialize can be detected.

There are more open research paths that will be explored later on. New stages can be incorporated to the pipeline, for example to divide and group tiles to fit the best size for the metrics and classifiers used, automatically. The multistage approach can also be adapted to other application domains like indoor scans or mobile mapping data, by changing the situations that can be identified in the second stage and changing the classifiers in the third stage.

Acknowledgements We wish to acknowledge the support received from the Centro de Investigación de Galicia "CITIC", funded by Xunta de Galicia and the European Union (European Regional Development Fund- Galicia 2014–2020 Program), by grant ED431G 2019/01. This research was funded by the Ministry of Science and Innovation of Spain (PID2019-104184RB-I00/AEI/10.13039/501100011033), by the

Galician Government under the Consolidation Program of Competitive Research Units (Ref. ED431C 2021/30) and by grant PID2022-136435NB-I00, funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", EU. Diego Teijeiro received financial support from Xunta de Galicia and the European Social Fund (ESF) of the European Union (predoctoral fellowship ref. ED481A-2019/231). The point clouds and LiDAR datasets used in this work belong to the LiDAR-PNOA data repository, region of Navarra, and were provided by Gobierno de Navarra. Dataset license: LiDAR-PNOA-cob2 2017 CC-BY 4.0 scene.es.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability Parts of the data that support the findings of this study are openly available in Centro Nacional de Información Geográfica, CNIG at <https://doi.org/10.7419/162.09.2020>, product ID LiDAR-PNOA-cob2, and were provided by Gobierno de Navarra.

Declarations

Conflict of interest No potential Conflict of interest was reported by the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Martín-García S, Balenović I, Jurjević L, Lizarralde I, Buján S, Ponce RA (2022) What is the most suitable height range of als point cloud and lidar metric for understory analysis? A study case in a mixed deciduous forest, Pokupsko Basin, Croatia. *Remote Sens* 14:2095. <https://doi.org/10.3390/rs14092095>
2. Doneus M, Höfle B, Kempf D, Daskalakis G, Shinoto M (2022) Human-in-the-loop development of spatially adaptive ground point filtering pipelines-an archaeological case study. *Archaeol Prospect* 29(4):503–524. <https://doi.org/10.1002/arp.1873>
3. Domingo D, van Vliet J, Hersperger AM (2023) Long-term changes in 3d urban form in four Spanish cities. *Landsc Urban Plan* 230:104624. <https://doi.org/10.1016/j.landurbplan.2022.104624>
4. Gazibara SB, Krkač M, Arbanas SM (2016) Landslide inventory mapping using lidar data in the city of zagreb (croatia). *J Maps* 15(2):773–779. <https://doi.org/10.1080/17445647.2019.1671906>
5. Wu Y, Sang M (2022) WeiWang: a novel ground filtering method for point clouds in a forestry area based on local minimum value and machine learning. *Appl Sci* 12:9113. <https://doi.org/10.3390/app12189113>
6. Qin N, Tan W, Ma L, Zhang D, Guan H, Li J (2023) Deep learning for filtering the ground fromals point clouds: a dataset, evaluations and issues. *ISPRS J Photogramm Remote Sens* 202:246–261. <https://doi.org/10.1016/j.isprsjprs.2023.06.005>
7. Sturari M, Paolanti M, Frontoni E, Mancini A, Zingaretti P (2017) Robotic platform for deep change detection for rail safety and security. In: 2017 European Conference on Mobile Robots (ECMR), pp 1–6. <https://doi.org/10.1109/ECMR.2017.8098668>
8. Yang B, Huang R, Li J, Tian M, Dai W, Zhong R (2017) Automated reconstruction of building lods from airborne lidar point clouds using an improved morphological scale space. *Remote Sens*. <https://doi.org/10.3390/rs9010014>

9. Xie Y, Tian J, Zhu XX (2020) Linking points with labels in 3d: a review of point cloud semantic segmentation. *IEEE Geosci Remote Sens Mag* 8(4):38–59. <https://doi.org/10.1109/MGRS.2019.2937630>
10. Meng X, Currit N, Zhao K (2010) Ground filtering algorithms for airborne lidar data: a review of critical issues. *Remote Sens* 2(3):833–860. <https://doi.org/10.3390/rs2030833>
11. Richter R, Behrens M, Döllner J (2013) Object class segmentation of massive 3d point clouds of urban areas using point cloud topology. *Int J Remote Sens* 34(23):8408–8424. <https://doi.org/10.1080/01431161.2013.838710>
12. Hu Z, Zhen M, Bai X, Fu H, Tai C-1 (2020) JSEnNet: joint semantic segmentation and edge detection network for 3D point clouds. In: *Computer vision—ECCV 2020*, pp 222–239
13. Dong Z, Yang B, Hu P, Scherer S (2018) An efficient global energy optimization approach for robust 3d plane segmentation of point clouds. *ISPRS J Photogramm Remote Sens* 137:112–133
14. Ge C, Du Q, Li W, Li Y, Sun W (2019) Hyperspectral and lidar data classification using kernel collaborative representation based residual fusion. *IEEE J Sel Top Appl Earth Obs Remote Sens* 12(6):1963–1973
15. Li Z, Zhang L, Tong X, Du B, Wang Y, Zhang L, Zhang Z, Liu H, Mei J, Xing X, Mathiopoulos PT (2016) A three-step approach for TLS point cloud classification. *IEEE Trans Geosci Remote Sens* 54(9):5412–5424
16. Chehata N, Guo L, Mallet C (2009) Airborne lidar feature selection for urban classification using random forests 38:207–212
17. Vosselman G, Coenen M, Rottensteiner F (2017) Contextual segment-based classification of airborne laser scanner data. *ISPRS J Photogramm Remote Sens* 128:354–371
18. Wang Z, Zhang L, Fang T, Mathiopoulos PT, Tong X, Qu H, Xiao Z, Li F, Chen D (2015) A multi-scale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. *IEEE Trans Geosci Remote Sens* 53(5):2409–2425
19. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS' 17*, Curran Associates Inc., Red Hook, NY, pp 3859–3869
20. Charles R, Su H, Kaichun M, Guibas LJ (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Los Alamitos, pp 77–85
21. Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: *Advances in neural information processing systems*, vol 30. [arXiv:1706.02413](https://arxiv.org/abs/1706.02413) [cs.CV]
22. Bouch A, Guerry J, Le Saux B, Audebert N (2018) Snapnet: 3d point cloud semantic labeling with 2D deep segmentation networks. *Comput Graph* 71:189–198
23. Meng H-Y, Gao L, Lai Y-K, Manocha D (2019) VV-Net: voxel VAE net with group convolutions for point cloud segmentation. In: *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp 8499–8507
24. Döllner J (2020) Geospatial artificial intelligence: potentials of machine learning for 3d point clouds and geospatial digital twins. *PFJ Photogramm Remote Sens Geoinf Sci* 88:15–24
25. Zhang J, Lin X, Ning X (2013) Svm-based classification of segmented airborne lidar point clouds in urban areas. *Remote Sens* 5(8):3749–3775
26. Wang A, He X, Ghamisi P, Chen Y (2018) Lidar data classification using morphological profiles and convolutional neural networks. *IEEE Geosci Remote Sens Lett* 15(5):774–778
27. Pingel TJ, Clarke KC, McBride WA (2013) An improved simple morphological filter for the terrain classification of airborne lidar data. *ISPRS J Photogramm Remote Sens* 77:21–30
28. Hui Z, Jin S, Xia Y, Nie Y, Xie X, Li N (2021) A mean shift segmentation morphological filter for airborne lidar dtm extraction under forest canopy. *Optics Laser Technol* 136:106728. <https://doi.org/10.1016/j.optlastec.2020.106728>
29. Maseda RC, Barrós DM (2012) LIDAR2MDTPlus Generación de Modelos Digitales de Terreno de pendiente variable a partir de datos LIDAR mediante filtro morfológico adaptativo y computación paralela sobre procesadores multinúcleo. Software registration ID: SC-102-12
30. Zhang W, Qi J, Peng W, Wang H, Xie D, Wang X, Yan G (2016) An easy-to-use airborne lidar data filtering method based on cloth simulation. *Remote Sens* 8:501
31. Hui Z, Li D, Jin S, Ziggah YY, Wang L, Hu Y (2019) Automatic dtm extraction from airborne lidar based on expectation-maximization. *Optics Laser Technol* 112:43–55. <https://doi.org/10.1016/j.optlastec.2018.10.051>

32. Axelsson P (2000) Dem generation from laser scanner data using adaptive tin models. *ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci* 33:110–117
33. Buján S, Sellers CA, Cordero M, Miranda D (2020) Dechpoints: a new tool for improving lidar data filtering in urban areas. *PFG J Photogramm Remote Sens Geoinf Sci* 88:239–255
34. Qin N, Tan W, Guan H, Wang L, Ma L, Tao P, Fatholahi S, Hu X, Li J (2023) Towards intelligent ground filtering of large-scale topographic point clouds: a comprehensive survey. *Int J Appl Earth Obs Geoinf* 125:103566. <https://doi.org/10.1016/j.jag.2023.103566>
35. Klápště P, Fogl M, Barták V, Gdulová K, Urban R, Moudrý V (2020) Sensitivity analysis of parameters and contrasting performance of ground filtering algorithms with uav photogrammetry-based and lidar point clouds. *Int J Digit Earth* 13(12):1672–1694
36. Aryal RR, Latifi H, Heurich M, Hahn M (2017) Impact of slope, aspect, and habitat-type on lidar-derived digital terrain models in a near natural, heterogeneous temperate forest. *PFG J Photogramm Remote Sens Geoinf Sci* 85:243–255
37. Deibe D, Amor M, Doallo R (2020) Big data geospatial processing for massive aerial lidar datasets. *Remote Sens* 12(4):719
38. Li Z, Hodgson ME, Li W (2018) A general-purpose framework for parallel processing of large-scale lidar data. *Int J Digit Earth* 11(1):26–47
39. Wang C, Hu F, Sha D, Han X (2017) Efficient LIDAR point cloud data managing and processing in a hadoop-based distributed framework. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci* 4:121–124
40. PNOA 2nd cover region of Navarra (2017) <https://filescartografia.navarra.es/>. Data source: Gobierno de Navarra. License: LiDAR-PNOA 2017 CC-BY 4.0 scne.es
41. LASer data format specification, version 1.4 r15 (2019) https://www.asprs.org/wp-content/uploads/2019/07/LAS_1_4_r15.pdf
42. Tucker CJ (1979) Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sens Environ* 8(2):127–150. [https://doi.org/10.1016/0034-4257\(79\)90013-0](https://doi.org/10.1016/0034-4257(79)90013-0)
43. Ground filtering as part of LAStools suite (2023) https://rapidlasso.de/lasground_new/. Accessed 12 April 2023
44. Sithole G, Vosselman G (2004) Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS J Photogramm Remote Sens* 59(1):85–101. <https://doi.org/10.1016/j.isprsjprs.2004.05.004>. Advanced Techniques for Analysis of Geo-spatial Data

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.