Exploitation of the Number of Return Echoes for DTM Extraction from Point Clouds Acquired by LiDAR UAS DJI Zenmuse L1

Francesca Matrone [1] *, Francesca Gallitto [1], Andrea Maria Lingua [1], Paolo Felice Maschio [1]

[1] Laboratory of Photogrammetry, Geomatics & GIS, Department of Environment, Land and Infrastructures Engineering (DIATI), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, (francesca.matrone, francesca.gallitto, paolo.maschio, andrea.lingua)@polito.it

Keywords: UAS, LiDAR, DTM extraction, return number, echoes, machine learning, Random Forest, morphological filter

Abstract

Following the enormous technological developments of LiDAR (Light Detection And Ranging) sensors, it is currently easier to find them commercially in the UA (Uncrewed Aerial Systems) sector. In particular, with the Zenmuse L1 by DJI (Dà-Jiāng Innovations) the market has grown globally, mainly due to the compactness of the product that is easily compatible with UAS. The L1 sensor can record up to three returns of the emanating signal, so it can acquire a larger amount of points, such as those below the vegetation. Therefore, in addition to the geometric information of the points, the Zenmuse L1 point clouds also provide other information, such as the number of echo returns from 1 to 3. This data could be exploited to improve the automatic extraction of the digital terrain model (DTM) from the point clouds, hopefully leading to the avoidance of manual correction. This research aims to focus on evaluating whether the addition of the return number feature can affect the identification of the ground points through different computational methods and can improve the time efficiency of state-of-the-art algorithms.

1. Introduction

The advent of Light Detection and Ranging (LiDAR) technology has revolutionized the way we perceive and understand landscapes. LiDAR systems capture vast amounts of data, generating highly detailed three-dimensional point clouds. In recent years, with the DJI solutions on the market, LiDAR systems have become very compact and affordable, guaranteeing an even higher level of detail thanks to the possibility of mounting it on a Unmanned Aerial Vehicles (UAV). The so-obtained point clouds serve as a rich source of information, invaluable for various applications ranging from urban planning to environmental monitoring.

However, within these dense point clouds lies a wealth of nuanced data that requires careful interpretation and extraction. One crucial aspect that significantly influences the accuracy and utility of derived products, such as Digital Terrain Models (DTMs), is the return number associated with each LiDAR point. Return numbers, or echoes, categorize individual points based on the number of times the laser pulse is reflected back to the sensor. These can range from first returns, representing the surface of objects like trees and buildings, to multiple returns, indicative of complex terrain features or vegetation layers. While all returns contribute to the overall point cloud, understanding and leveraging the return numbers could be paramount, particularly in DTM extraction processes, since DTMs need to be devoid of above-ground features such as vegetation and structures. Accurate DTMs are essential for applications like flood modelling (Alho et al., 2009), viewshed analysis (Maloy et al., 2001), and slope stability assessment (Subramaniam et al., 2022). The precision of these models hinges on the ability to distinguish between ground points and non-ground points within the LiDAR point cloud.

In this context, the utilization of return numbers emerges as a critical factor in refining DTM extraction methodologies. By differentiating between first returns, which predominantly represent ground surfaces, and subsequent returns, which may signify vegetation or man-made structures, it is possible to enhance the accuracy of terrain modelling processes. Leveraging the information encoded in return numbers enables the development of robust algorithms tailored to isolate ground points with greater precision, thus yielding more accurate and reliable DTMs (Chen et al., 2017; Hyyppä et al., 2005; Štroner et al., 2023).

In particular, in this contribution, we would like to investigate, on one side, if it is possible to use the return number to improve the computing time of state-of-the-art morphological algorithms, on the other hand if it could be considered effective for the point cloud semantic segmentation through Machine Learning (ML) algorithms such as the Random Forest.

This investigation was carried out on the data derived from the previously mentioned DJI Zenmuse L1 sensor, which has as one of the most interesting features, the integration between the LiDAR, the RGB camera and the IMU (Inertial Measurement Unit) module in one single hardware mounted on a 3-axis stabilized gimbal. Table 1 shows some specifications (Teppati Losè et al., 2022).

LiDAR sensor	
Range	450 m @ 80% reflectivity, 0 klx 190 m @ 10% reflectivity, 100 klx
Point Rate	Single return: max. 240,000 pts/s Multiple return: max. 480,000 pts/s
Positional accuracy	Horizontal: 10 cm @ 50 m Vertical: 5 cm @ 50 m
Point Cloud Coloring	Reflectivity, Height, Distance, RGB
Ranging Accuracy	3 cm @ 100 m
FoV	Non-repetitive scanning pattern: 70.4° (horizontal) × 77.2° (vertical) Repetitive scanning pattern: 70.4° (horizontal) × 4.5° (vertical)
Yaw Accuracy	Real-time: 0.3°, Post-processing: 0.15°
Pitch / Roll Accuracy	Real-time: 0.05°, Post-processing: 0.025°
RGB sensor	
RGB camera	1” CMOS 20 MP
Focal Length	8.8 mm / 24 mm (Equivalent)

Table 1. Specifications of DJI Zenmuse L1.

1.1 The case study

Three different methodologies have been applied to evaluate whether the addition of the return number information improves the discrimination of the ground points. Each methodology is based on a different classification strategy and focused on investigating different insights.

The case study depicts a mountain area in the west Italian alpine region with slopes and different kinds of patterns where small and medium-sized shrubs alternate with large areas of both flat and noisy ground (Figure 1).

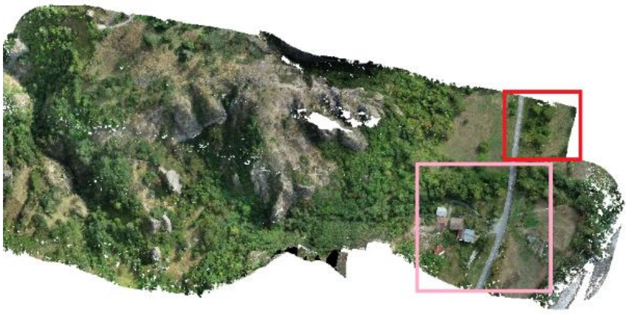


Figure 1. Overall point cloud with the subset of points used for the tests (red for Sample 1, pink for the Sample 2).

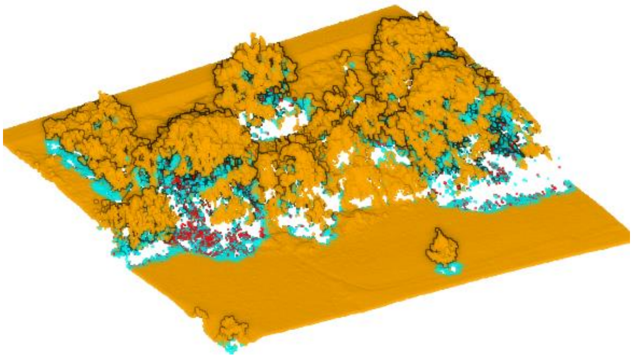
A small testbed area was selected out of the whole point cloud for the preliminary tests and to set up and validate the algorithms. It included only three categories (ground, road and vegetation) and it counted 651.898 points out of 53.303.310; we will refer to it as Sample 1 (Figure 2).

A second wider sample (Sample 2) was then selected, with 8.679.442 points and additional categories such as buildings and electricity poles (Figure 3). Both the point clouds had the standard features directly derived from the Zenmuse L1 LiDAR: RGB, Scan angle rank, GPS time, number of returns and intensity.

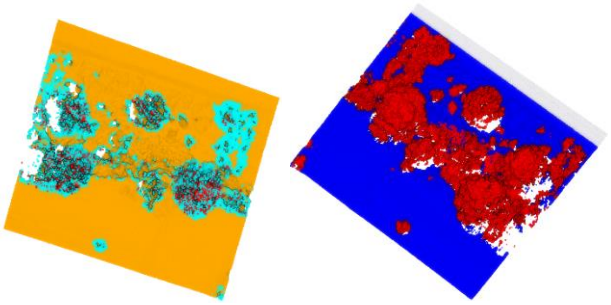
Table 2 shows the main features of the two samples.

	Features	Numbers of returns	Numbers of points
Sample 1 (S1)	RGB, number of returns, Scan angle rank, GPS time	3	617.001
Sample 2 (S2)			8.679.442

Table 2. Main features of the two samples.

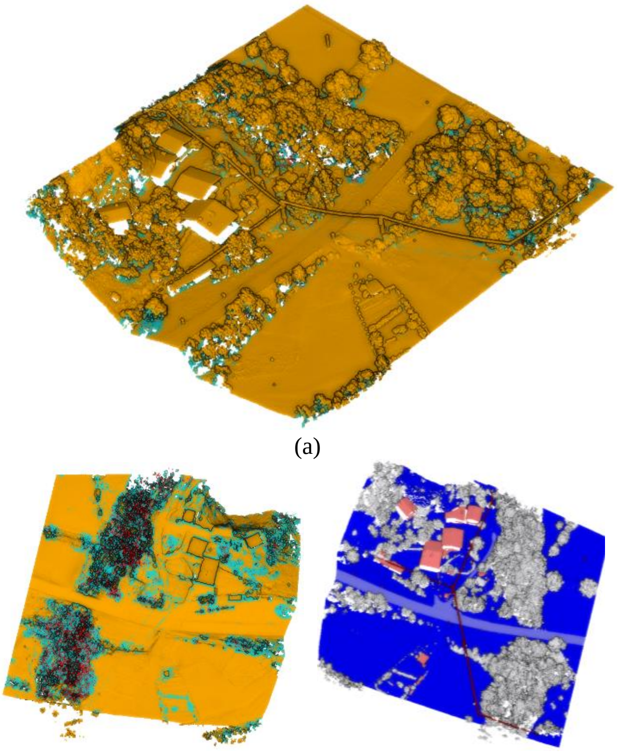


(a)



(b) (c)

Figure 2. Sample 2 (S2), with scalar field based on the number of returns: 1 orange, 2 light blue, 3 red. Top view (a) and bottom view (b). Manually classified point cloud, top view (c): ground – blue, road – grey, vegetation – red.



(a)

(b) (c)

Figure 3. Sample 2 (S2), with scalar field based on the number of returns: 1 orange, 2 light blue, 3 red. Top view (a) and bottom view (b). Manually classified point cloud, top view (c): ground – blue, road – violet, vegetation – grey, buildings – pink, power lines – red.

2. Methodology and results

Three algorithms were tested, each of which defined a different method and approach to investigate whether the information given by the return number could be useful to extract the ground points. These methods were applied to both S1 and S2, and a confusion matrix was calculated to evaluate the performances. In addition, the processing time was also recorded to compare the timing. Methods 1 and 2 were developed in MATLAB, involved the geometric structure of the point clouds and were compared to assess the effectiveness of Method 2, while Method 3 consists of the Random Forest ML classifier.

2.1 Method 1: simple morphological filter algorithm

The first method to be tested (that constituted the reference one) consisted of a simple morphological filter (SMRF) algorithm, which extracts the ground points by evaluating only the geometry of each individual point (Pingel et al., 2013). In this case, return number information is not used. The algorithm consists of three steps: 1) creating a minimum elevation surface map from the point cloud data; 2) segmenting the surface map into ground and off-ground grid elements; 3) segmenting the original point cloud data and subsampling it with a grid-spaced approach. To create a minimum elevation surface map, the process begins by dividing the point cloud data into grids along the xy-plane, the grid size is specified.

Subsequently, for each grid element (pixel), the lowest elevation value (*Zmin*) is found. To generate a segment surface map, the process is initiated by employing a morphological opening operation on the minimum surface map. This operation entails the application of both an erosion filter and a dilation filter consecutively. The structuring element's shape and its window radius play a crucial role in defining the search neighbourhood for this operation. Typically, a disc-shaped structuring element is utilised, commencing with a window radius of 1 pixel.

Following this, the slope between the minimum surface and the opened surface maps is computed at each grid element. The pixel is categorised as non-ground if the disparity surpasses the elevation threshold. These steps are carried out iteratively, progressively increasing the window radius by 1 pixel in each iteration until it reaches the specified maximum radius.

The process unfolds as follows to segment the original point cloud: initially, the binary mask generated earlier is employed on the original minimum surface map to filter out non-ground grids. Subsequently, the vacant grids are filled using image interpolation techniques to construct an estimated elevation model. Then, the disparity in elevation between every point in the original point cloud and its corresponding point in the estimated elevation model is computed. Pixels with discrepancies exceeding the threshold are classified as non-ground.

Moreover, the incline of the elevation model at each point is scaled and added to the threshold value. This step aids in pinpointing ground points on steep slopes. Through this iterative process, a binary mask is derived where each pixel of the point cloud undergoes classification as either ground or non-ground based on the segmentation criteria.

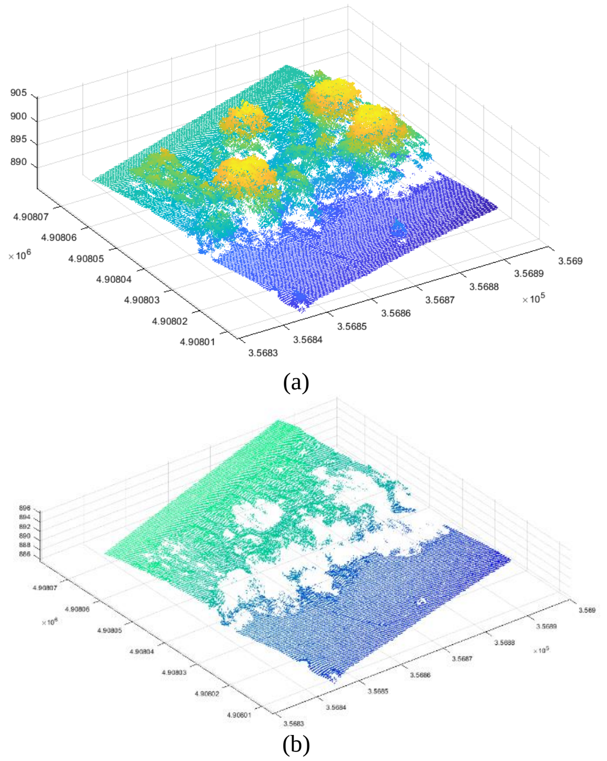
The algorithm has already been implemented in the function *segmentGroundSMRF(..)* since MATLAB version R2021a. The argument to be inserted into the function is the point cloud, while the result is a logical vector, where the value 0 represents off-ground points while 1 represents the ground points.

The MATLAB code is represented below:

```
groundPtsIdx = segmentGroundSMRF(ptCloud);
ptCloudWithGround = select(ptCloud,groundPtsIdx);
nonGroundPtCloud = select(ptCloud, ~groundPtsIdx);
```

2.1.1 Sample 1: For the smallest area, the code executed quite fastly, totalling 8 seconds, and the result was considered satisfactory since higher than 98% (Figure 4).

The next step was to compare the result with a manual classification of the point cloud; the results are shown in the confusion matrix $CM_{M1,S1\ (Method1,\ Sample1)}$ Table 3).



(a)

(b)

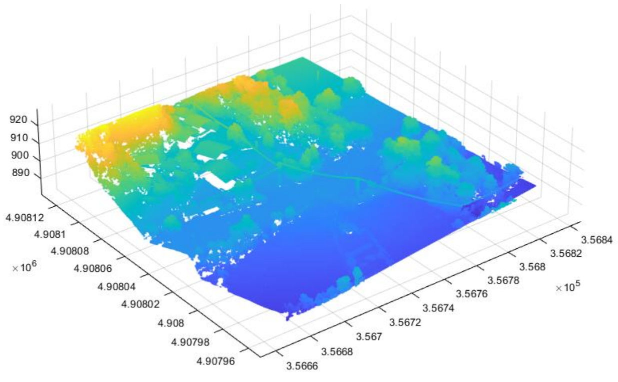
Figure 4. Sample 1, visualization of the point cloud according to the z-value scalar field (a), result of method 1 (b).

$$CM_{M1,SC} = \begin{pmatrix} 405623 & 5359 \\ 5772 & 200247 \end{pmatrix}$$

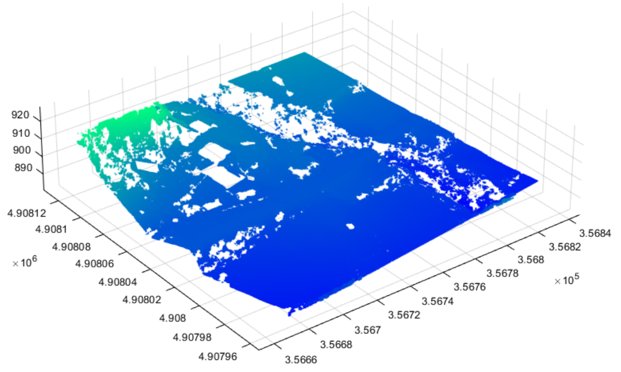
Metrics	Value
Accuracy	98.19 %
Precision	98.69 %
Recall	98.59 %
F1	98.64 %

Table 3. Results of Method 1 – Sample 1

2.1.2 Sample 2: As for S2 (13 times larger than the S1), the code processes fairly quickly for a total of 14 seconds. The result showed still high performances with an overall accuracy of 95%, and with buildings and urban elements properly removed as well as the vegetation (Figure 5). As for S1 a comparison with manual classification was performed, these results are contained in the confusion matrix $CM_{M1,S2}$ (Table 4).



(a)



(b)

Figure 5. Sample 2, visualization of the point cloud according to the z-value scalar field (a), result of method 1 (b).

$$CM_{M1,MC} = \begin{pmatrix} 5128968 & 327684 \\ 57551 & 3165241 \end{pmatrix}$$

Metrics	Value
Accuracy	95.56 %
Precision	93.99 %
Recall	98.89 %
F1	96.38 %

Table 4. Results of Method 1 – Sample 2

2.2 Method 2: geometry-based approach with return number embedding

The second method is very similar to the previous one, but it adds the return number to investigate the possibility of making Method 1 even faster. In particular, a mask of size *n x n* is created and iterated through the point cloud as a kernel; then, for each mask, the smallest *z value* among all the points is stored. All the points that exceed a certain threshold from the minimum *z* are classified as off-ground. Unlike the previous method, the return numbers data has been partially used: in fact, since the bare terrain areas are mainly characterized by a return number of 1 (as depicted in Figure 2 and 3), the method bypasses kernels with only 1 values, directly associating the ground class.

By taking advantage of this feature it is possible to make the first method faster and more efficient, keeping also a higher number of points. However, the potential of the echoes is still not fully addressed and evaluated, thus the third method comes to help. Several tests were performed to find the best combination of the kernel size and the minimum z-threshold; the values to get the best result without a high increasing of the processing time are 5 meters x 5 meters for the mask and 2 meters for the z-threshold. The MATLAB code for method 2 is represented below:

```
while MXmin < xmax
while MYmin < ymax
num_punti_maschera=0;
for i=1:N
if x(i)> MXmin && x(i)<MXmax && y(i)>MYmin &&
y(i)<MYmax %condition inside mask

num_punti_maschera=num_punti_maschera+1; %how many
points
ZM(num_punti_maschera,1)=z(i);%save xyz
XM(num_punti_maschera,1)=x(i);
YM(num_punti_maschera,1)=y(i);
NMM(num_punti_maschera,1)=returnNumb(i);
indice(num_punti_maschera,1)=i;
end
end
if num_punti_maschera>0 %if we have points inside
mask
[z_min,I]=min(ZM); %must to be clean
NMM_max=max(NMM);
if NMM_max~=1 %if we have just 1 of returns number,
is ground
for k=1:num_punti_maschera
if ZM(k,1) > z_min + delta_z
ground(indice(k,1),1)=0; %not ground
end
end
end
clear All parametersZM
end
MYmin=MYmin+delta; %update y mask
MYmax=MYmax+delta;
end
MXmin=MXmin+delta; %update x, y of mask
MXmax=MXmax+delta;
MYmin=ymin;
MYmax=ymin+delta;
End
```

2.2.1 Method 2 – Sample 1: as hypothesized Method 2 turns out to be faster, for S1 it processed in 1.3 seconds, almost 8 times faster than Method 1. The result (Figures 6) was compared with manual classification, and the confusion matrix $CM_{M2,S1}$ and metrics are reported in Table 5.

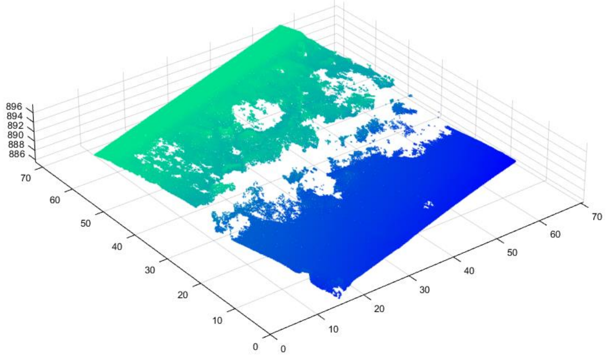


Figure 6. Sample 1, result of method 2.

$$CM_{M2,SC} = \begin{pmatrix} 400124 & 5070 \\ 10150 & 235412 \end{pmatrix}$$

Metrics	Value
Accuracy	97.66%
Precision	98.75%
Recall	97.53%
F1	98.13%

Table 5. Results of Method 2 – Sample 1.

2.2.2 Method 2 – Sample 2: in this case, the elapsed time is 62 seconds, way higher than the one obtained with the first method. The result is depicted in Figure 7, and the overall metrics are presented in Table 6.

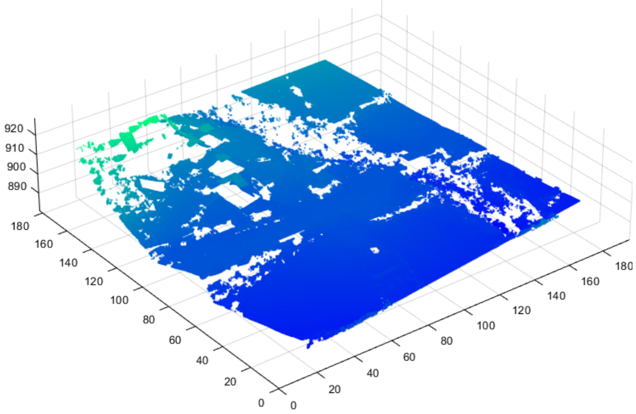


Figure 7. Sample 2, result of method 2.

$$CM_{M2,MC} = \begin{pmatrix} 5059434 & 310012 \\ 48789 & 3261213 \end{pmatrix}$$

Metric	Value
Accuracy	95.87 %
Precision	94.23 %
Recall	99.04 %
F1	96.58 %

Table 6. Results of Method 2 – Sample 2.

2.3 Method 3: Random Forest

The following approach investigated the use of machine learning algorithms, such as the Random Forest, to define if the addition of the echoes data does really affect the final classification (Wang et al. 2019).

In particular, a set of three tests was carried out for both S1 and S2, while for S2, additional tests were performed. For all the tests, various combinations between the splitting criteria to measure the impurity of the nodes (gini index or entropy), number of trees (100, 150 or 200) and scalers (none, scaler 1 or scaler 2) have been conducted, while the depth of the tree was always set to *none*. In the following sections, only the best results are reported.

As regards the scalers, they have been introduced to better address the presence of 3D features (Weinmann et al., 2015) and they consist in scaler1 which standardizes features by removing the mean and scaling to unit variance, and scaler 2 that scales features using statistics that are robust to outliers (Matrone et al. 2021).

In particular, the considered 3D features have been (Figure 8):

- RGB – radiometric component
- Return number (*feature 1 – f1*)
- Scan angle rank (*f2*)
- GPS time (*f3*)
- Verticality (radius 0.6 m) (*f4*)
- Planarity (radius 0.6 m) (*f5*)
- Eigenentropy (radius 0.4 m) (*f6*)
- Surface variation (radius 0.6 m) (*f7*)

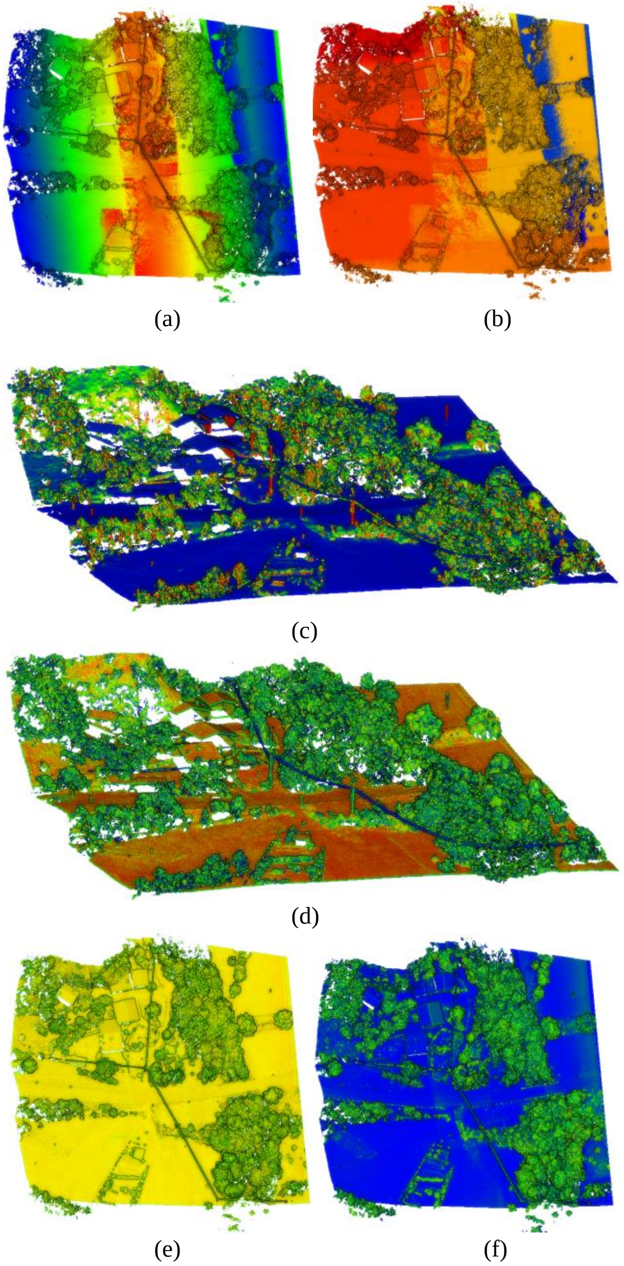


Figure 8. Scan Angle Rank (a), GPS time (b), Verticality (c), Planarity (d), Eigentropy (e), Surface variation (f).

Table 7 contains the configurations of the features selected for the various tests.

The third configuration (C3) was added to test the point cloud as restituted by the Zenmuse L1. The fifth configuration (C5), on the other hand, was investigated as a consequence of the results obtained from C3 (see § 2.3.1).

The scenes were then divided into training, validation and test, in order to have the presence of all the classes in each dataset (Figure 9)

Configuration		S1	S2
C1	RGB	yes	yes
C2	RGB + f1	yes	yes
C3	RGB + f1, f2, f3	no	yes
C4	RGB + f1, f2, f3, f4 f5, f6, f7	yes	yes
C5	RGB + f1, f4 f5, f6, f7	no	yes

Table 7. Configuration of the feature selection for the various tests performed.

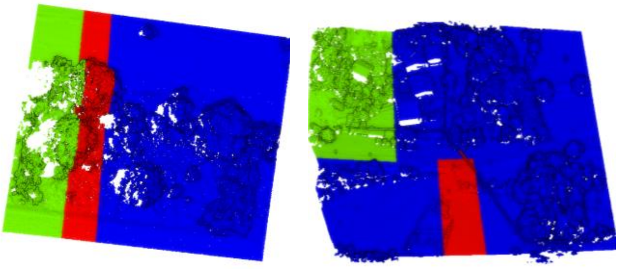


Figure 9. Dataset divided into: training – green, validation – red, test – blue. Sample 1 (a) and Sample 2 (b).

2.3.1 Method 3 – Sample 1 (S1):

The first set of tests involved the smaller point cloud with only three classes. The experiments were carried out on Google Colaboratory, and the training + test phases took 49 seconds for the configuration with only the radiometric components (C1 of Table 7), 50 seconds for the combination with the addition of the echoes (C2) and 1 minute and 11 seconds with all the features (C5). Tables 8, 9 and 10 report the obtained results, while Figure 10 shows a visual comparison between the three predictions and the ground truth.

	Precision	Recall	F1-score
Ground	0.83	0.69	0.75
Road	1	0.95	0.97
Vegetation	0.53	0.71	0.6
		Overall accuracy	0.71
Weighted avg	0.74	0.71	0.72

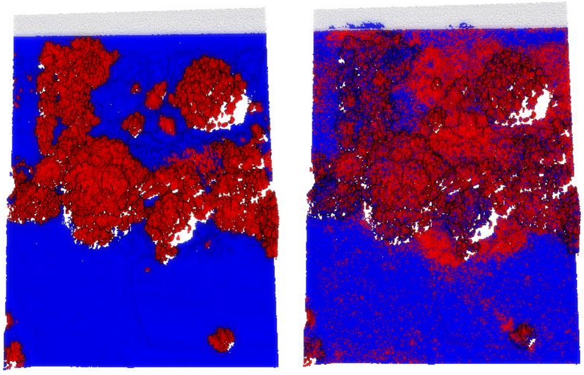
Table 8. Results of the test with only the RGB as input features (C1).

	Precision	Recall	F1-score
Ground	0.87	0.77	0.82
Road	1	0.96	0.98
Vegetation	0.62	0.78	0.69
		Overall accuracy	0.78
Weighted avg	0.8	0.78	0.79

Table 9. Results of the test with the RGB + return number as input features (C2).

	Precision	Recall	F1-score
Ground	0.89	0.91	0.9
Road	1	0.9	0.95
Vegetation	0.81	0.79	0.8
		Overall accuracy	0.87
Weighted avg	0.87	0.87	0.87

Table 10. Results of the test with all the features as input (C4).



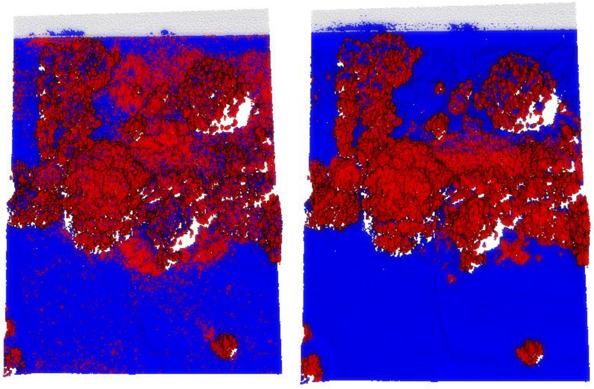


Figure 10. Ground truth (a) and predictions: based on C1 - only RGB (b), C2 – RGB + return number (c), C4 – all the features (d).

Comparing the results, it is possible to highlight that the introduction of the information related to the return number helps improve the overall metrics, leading to an increase of about 7% for the overall accuracy and 6% for the F1 Score. In particular, the ground class achieved 82% of the F1 Score compared to the previous 75%. These results are further improved with the addition of other features (Table 10), respectively +7% and +8%, confirming the positive trend demonstrated in other previous contributions (Matrone et al., 2020).

However, it was necessary to understand whether the improvement in performance was due solely to the introduction of new features, respecting the abovementioned trend, or whether the return numbers provided effective and relevant informative content for the algorithms.

In this regard, we added a further test, namely configuration 5 (C5) with all the features except for the Scan angle rank and GPS time. This choice was dictated by the fact that visually these two features seemed not to add significant information and not to be able to discriminate any particular category.

Finally, a further configuration (C3) was envisaged to understand if and to what extent the data extracted as such from sensors such as the LiDAR L1 can be effective in predictions and classifications for the generation of DTMs. These tests were conducted only for Sample 2.

2.3.2 Method 3 – Sample 2 (S2):

Below are report the results of the five configurations (Table 11, 12, 13, 14, 15 and 16) and the visual predictions (Figure 11).

In this case, the experiments were also carried out on Google Colaboratory, and the training + test phases took 14 to 17 minutes.

	Precision	Recall	F1-score
Ground	0.72	0.77	0.74
Road	0.89	0.49	0.63
Vegetation	0.71	0.63	0.67
Buildings	0.39	0.78	0.52
Power lines	0.02	0	0
		Overall accuracy	0.70
Weighted avg	0.71	0.7	0.7

Table 11. Results of the test with only the RGB as input features (C1).

	Precision	Recall	F1-score
Ground	0.76	0.82	0.79
Road	0.87	0.53	0.66
Vegetation	0.78	0.7	0.73
Buildings	0.43	0.74	0.55
Power lines	0.2	0.02	0.04
		Overall accuracy	0.76
Weighted avg	0.76	0.76	0.76

Table 12. Results of the test with the RGB + return number as input features (C2).

	Precision	Recall	F1-score
Ground	0.75	0.8	0.77
Road	0.7	0.61	0.65
Vegetation	0.74	0.72	0.73
Buildings	0.24	0.09	0.13
Power lines	0.45	0.2	0.28
		Overall accuracy	0.74
Weighted avg	0.73	0.74	0.73

Table 13. Results of the test with the L1 features as input (C3).

	Precision	Recall	F1-score
Ground	0.78	0.95	0.85
Road	0.95	0.62	0.75
Vegetation	0.92	0.71	0.8
Buildings	0.64	0.47	0.54
Power lines	0.7	0.04	0.07
		Overall accuracy	0.82
Weighted avg	0.84	0.82	0.82

Table 14. Results of the test with all the features as input (C4). Gini index, scaler 1.

	Precision	Recall	F1-score
Ground	0.79	0.95	0.86
Road	0.93	0.6	0.73
Vegetation	0.92	0.72	0.81
Buildings	0.7	0.6	0.65
Power lines	0.72	0.02	0.05
		Overall accuracy	0.83
Weighted avg	0.85	0.83	0.83

Table 15. Results of the test with all the features as input (C4). Entropy, scaler 1.

	Precision	Recall	F1-score
Ground	0.8	0.95	0.87
Road	0.92	0.62	0.74
Vegetation	0.93	0.75	0.83
Buildings	0.73	0.67	0.7
Power lines	0.41	0.02	0.03
		Overall accuracy	0.85
Weighted avg	0.86	0.85	0.84

Table 16. Results of the test with all the features as input except for the Scan angle rank (*f2*) and GPS time (*f3*) (C5).

From these metrics, it is clear that the behaviour is similar to Sample 1 if considering C1, C2, and C4, thus validating the previous conclusion; but, above all, the C5 test shows that the addition of features is not always a positive factor. In fact, in this case, the elimination of two features that could not properly describe the classes led to slightly higher metrics in the predictions, e.g. overall accuracy of 85% (C5) versus 82/83% (C4), and for the ground category, an increasing of about 2%.

Finally, regarding the comparison between the C2 and C3 configuration, it is highlighted that the data directly extracted from the L1 is not optimal for classification with ML algorithms such as Random Forest, but they need a pre-analysis or filtering of the features.

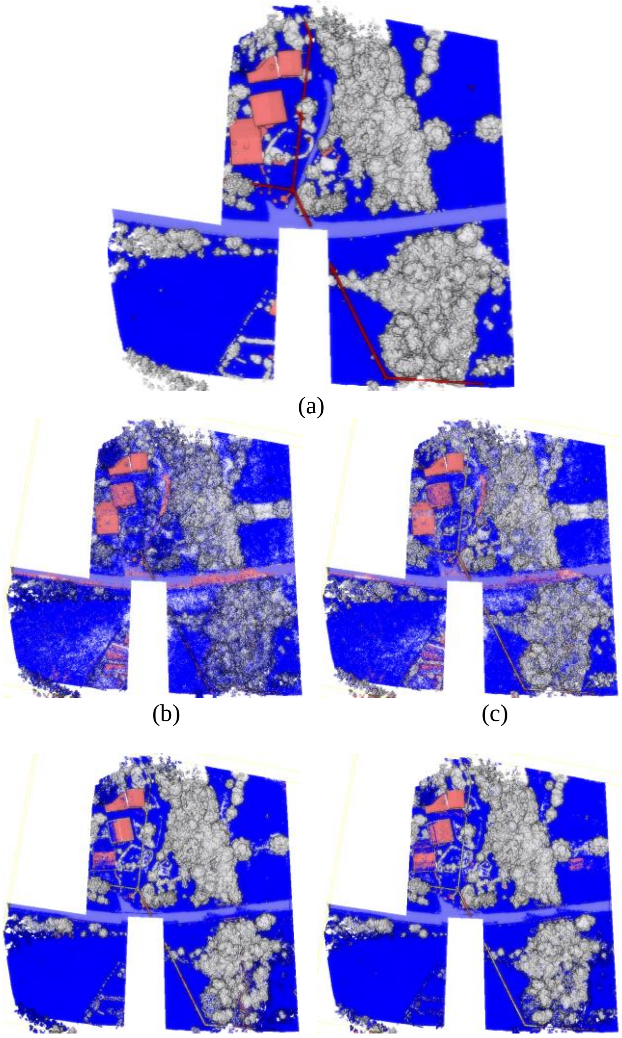


Figure 11. Ground truth (a) and predictions: based on C1 - only RGB (b), C2 – RGB + return number (c), C4 – all the features (d), C5 – all the features except for the Scan angle rank (*f2*) and GPS time (*f3*).

3. Conclusions

In this contribution, the exploitation of the return number from the DJI Zenmuse L1 for DTM extraction has been investigated. Preliminary tests on a MATLAB state-of-the-art morphological filter algorithm were conducted to enhance its computing time.

This method (method 1) showed promising results in terms of accuracy and precision but has higher processing times for Sample 1, while it performs better in Sample 2. On the other hand, the proposed method 2 is more performant and precise in the point cloud where only vegetation is present (Sample 1), while in Sample 2, where we find building roofs and urban elements, there is a consistent misclassification with the terrain category. The results shown for this proposed method are only preliminary and further investigations and in-depth studies and tests should be carried out. In addition, a limitation of this method consists of the fact that the return number is only used to initially filter the ground points, on one side making an assumption that is not fully generalizable, on the other side restraining the intrinsic potentialities of this kind of data.

As regards the use of the Random Forest, it emerged that the return number is effectively a useful feature in discriminating the ground class for DTM extraction, even if its coupling with additional features is desirable anyway.

A further development of this research could lie in the ability to understand not *if*, but *how* the number of returns influences the final predictions, analysing the possible correlations between the echoes and the classes to be discriminated.

Acknowledgements

This study is carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors’ views and opinions; neither the European Union nor the European Commission can be considered responsible for them.

References

Alho, P., Hyyppä, H. and Hyyppä, J., 2009. Consequence of DTM precision for flood hazard mapping: A case study in SW Finland. Nordic journal of surveying and real estate research, 6(1).

Axelsson, P., 1999. Processing of laser scanner data—algorithms and applications, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 54, Issues 2–3, pp. 138-147.

Chen, Z., Gao, B. and Devereux, B., 2017. State-of-the-art: DTM generation using airborne LIDAR data. Sensors, 17(1), p.150.

Hyyppä, H., Yu, X., Hyyppä, J., Kaartinen, H., Kaasalainen, S., Honkavaara, E. and Rönnholm, P., 2005. Factors affecting the quality of DTM generation in forested areas. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(3/W19), pp.85-90.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 53, Issue 4, pp. 193-203.

Maloy, M.A. and Dean, D.J., 2001. An accuracy assessment of various GIS-based viewshed delineation techniques. Photogrammetric engineering and remote sensing, 67(11), pp.1293-1298.

Matrone, F., Grilli, E., Martini, M., Paolanti, M., Pierdicca, R. and Remondino, F., 2020. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. ISPRS International Journal of Geo-Information, 9(9), p.535. https://doi.org/10.3390/ijgi9090535

Pingel, T.J., Clarke, K.C. and McBride, W.A., 2013. An improved simple morphological filter for the terrain classification of airborne LIDAR data. ISPRS journal of photogrammetry and remote sensing, 77, pp. 21-30.

Pirotti, F., Guarnieri, A. and Vettore, A., 2013. Ground filtering and vegetation mapping using multi-return terrestrial laser scanning. ISPRS Journal of Photogrammetry and Remote Sensing, 76, pp.56-63.

Russell G. Congalton, 1991. A review of assessing the accuracy of classifications of remotely sensed data, Remote Sensing of Environment, Volume 37, Issue 1, pp. 35-46.

Silván-Cárdenas, J.L., Wang, J.L., 2006. A multi-resolution approach for filtering LiDAR altimetry data, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 61, Issue 1, pp. 11-22.

Štroner, M., Urban, R., Křemen, T. and Braun, J., 2023. UAV DTM acquisition in a forested area–comparison of low-cost photogrammetry (DJI Zenmuse P1) and LiDAR solutions (DJI Zenmuse L1). European Journal of Remote Sensing, 56(1), p.2179942.

Subramaniam, T.R., Suhaimi, N.N.A., Paizol, A. and Nor, A.H.M., 2022. Determination of slope stability using (uav) unmanned aerial vehicle. Multidisciplinary Applied Research and Innovation, 3(2), pp.292-301.

Teppati Losè, L., Matrone, F., Chiabrando, F., Giulio Tonolo, F., Lingua, A. and Maschio, P., 2022. New developments in lidar uas surveys. performance analyses and validation of the Dji zenmuse L1. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, pp.415-422. https://doi.org/10.5194/isprs-archives-XLIII-B1-2022-415-2022

Weinmann, M., Jutzi, B., Hinz, S. and Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 105, pp.286-304.

Wang, C., Shu, Q., Wang, X., Guo, B., Liu, P. and Li, Q., 2019. A random forest classifier based on pixel comparison features for urban LiDAR data. ISPRS Journal of Photogrammetry and Remote Sensing, 148, pp.75-86.