New Efficient and Automatic Approach to Extract Dendrometric Features from Terrestrial LiDAR Point Clouds in Forest Inventories

Fernando J. Aguilar[1](✉), Abderrahim Nemmaoui[1], Flor Álvarez-Taboada[2], Francisco A. Rodríguez[1], and Manuel A. Aguilar[1]

[1] Department of Engineering/CIAIMBITAL Research Center, University of Almería, Carretera de Sacramento S/N, 04120 Almería, Spain
faguilar@ual.es

[2] Department of Mining Technology, Topography and Structures, University of León, 24404 Ponferrada, Spain

1 Introduction

Forest ecosystems provide numerous services and social benefits, including carbon storage, habitat for fauna, recreation, and the provision of wood and non-wood products [1]. This positive impact of forest ecosystems is especially relevant in the Mediterranean basin, where they are considered the most important ecosystems, being widely recognized as the main providers of ecosystem services.

Despite the above, there is an alarming lack of forest monitoring data, which is often an obstacle in decision-making regarding management activities and protection of forests by public and private entities.

During the last decade, both terrestrial laser scanning (TLS) and portable terrestrial LiDAR (PTL) have emerged as a technology that allows digitizing forests at the centimeter level. While TLS are stationary systems that must be fixed on a tripod, PTL devices use a simultaneous localization and mapping (SLAM) algorithm to reference laser distance measurements in 3D space while the device is in motion and without the need for of a global navigation satellite system (GNSS). The TLS and PTL point clouds enable fine-scale forest management, being considered as a completely new way to face forest inventories [2]. However, some barriers need to be overcome to efficiently and automatically transform scattered point clouds into meaningful information for forest management.

With respect to what meaningful information is potentially extractable from point clouds, many approaches have focused on dendrometric characterization [3]. This tree-centric approach requires dealing with the development of algorithms aim at extracting single tree features (i.e., tree location, DBH (diameter at breast height) and tree height). This issue can be addressed by three main approaches [4]: (i) Individual Tree Detection (ITD) algorithms based on direct detection of single tree features, (ii) Individual Tree Segmentation (ITS) algorithms more focused on segmenting trees as a first step for

C. Manchado del Val et al. (Eds.): INGEGRAF 2023, LNME, pp. 330–341, 2024.
https://doi.org/10.1007/978-3-031-51623-8_32

the subsequent extraction of dendrometric variables, and iii) Individual Tree Crown Delineation (ITCD) methods designed to delineate crown boundaries.

Although detecting trees and estimating their DBH and height has attracted significant efforts during the last decade [3], providing automatic tree-level forest inventories from very dense point clouds remains a challenge. This is due to different issues, such as point cloud noise, shrubs and understory vegetation hiding stems, laser beam occlusions caused by overlapping trees, and the highly diverse structure and morphology of trees [4]. Furthermore, many current approaches are computationally demanding and have an excessive number of parameters that the user needs to tune to tailor the feature extraction algorithm to each forest.

Taking into account the above-mentioned background, this work aims at introducing and assessing the performance of UALtree, a software tool written in Matlab® which makes use of the combination of ITS and ITD approaches to detect trees and estimate their position, height and DBH without having to deal with setting an excessive number of tuning parameters. The results obtained from UALtree were compared to two state-of-the-art algorithms such as AID-FOREST [5] and 3D FOREST [6].

2 Materials and Methods

2.1 Study Sites and Field Data Collection

The three aforementioned methods were tested in nine forest plots. Six of them were located in Sierra de María-Los Vélez Natural Park (Almeria, Spain) (Fig. 1). These six plots contain reforested stands of Aleppo pine (*Pinus halepensis* Mill.) with variable density, tree height and presence of shrubs and low vegetation (Table 1). These six plots were square with a side of 25 m. The fieldwork campaign was conducted in July 2021, using a Faro™ Focus3D X-330 TLS. Five scan positions were evenly distributed in each plot within a radius of 12.5 m from the center, one located close to the center of the plot and the remaining four located near the middle of the four sides, following the method described in Nemmaoui et al. [7].

The three remaining plots (Toral Vados, Villamañán and Villasabariego in Table 1) were rectangular shape plots with an average side of 31 m located in three different poplar plantations composed of several clones (Populus × interamericana and Populus × euroamericana) located in the province of León (Spain). Toral Vados is a 5 × 5 m plantation of the Unal clone 8-year-old, while Villamañán is a 12-year-old multiclonal plantation planted at 6x6 m and pruned. Finally, Villasabariego is also made up of several poplar clones planted at 6x6 m and pruned, with an age of 11 years. The three poplar plots were scanned in October 2022 with a PTL GeoSLAM™ ZEB Horizon scanner that worked without georeferencing (relative coordinates). The PTL trajectories followed the streets of the plantations trying to minimize occlusions and ensuring a closed loop to minimize SLAM drift.

The most important dasometric characteristics of the nine reference plots are shown in Table 1. It includes the PH350 index, a plot-level uniformity index that estimates the existing variability in tree growth rates. Among the Aleppo pine plots, only the plots 10B1P and 10B2P can be considered relatively homogeneous (PH350 in the range of 0.37 and 0.50) (Table 1), while practically all poplar plots are.

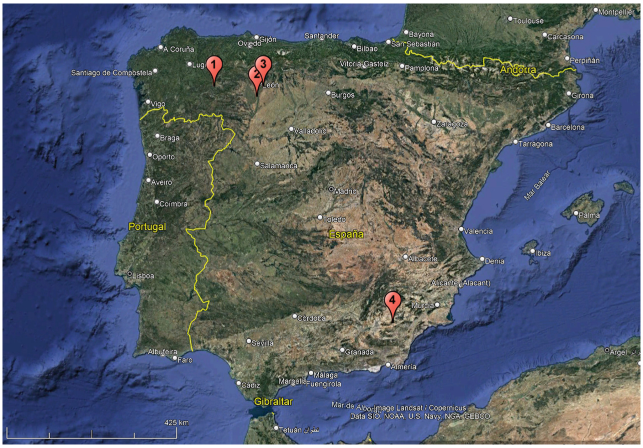

Fig. 1. Location of the study sites. 1) Poplar plot located in Toral de los Vados (León). 2) Poplar plot located in Villamañán (León). 3) Poplar plot located in Villasabariego (León). 4) Six Aleppo pine plots located in Sierra María-Los Vélez Natural Park (Almería).

Table 1. Dasometric characteristics. Number of trees (N), tree density (D), vegetation cover (VC), plot-level uniformity index ($PH^3 50$), basal area (G), Lorey´s height (Lh), and mean slope (M_{slope}).

Reference plot	Kind of tree	N	D (trees/ha)	VC (%)	$PH^3 50$	G (m^2/ha)	Lh (m)	Mslope (°)
18A1P	Aleppo pine	34	544	55.17	0.30	26.31	7.45	12.77
16A2P	Aleppo pine	28	448	45.4	0.30	15.33	6.65	15.92
13A3P	Aleppo pine	21	336	37.32	0.27	10.55	6.68	12.63
13A1P	Aleppo pine	23	368	40.57	0.24	8.28	5.55	16.55
10B2P	Aleppo pine	8	128	32.61	0.39	14.4	12.03	12.45
10B1P	Aleppo pine	18	288	51.69	0.38	24.32	10.57	9.61
Toral Vados	Poplar	33	326	64.63	0.42	14.63	22.76	1.43
Villamañán	Poplar	34	285	76.48	0.47	19.10	24.56	1.42
Villasabariego	Poplar	34	298	47.57	0.36	10.43	20.42	1.13

The Aleppo pine fieldwork was carried out between the end of May and the beginning of June 2021. Tree position and DBH was registered according to the methods described in Nemmaoui et al. [7], while tree height was measured on a high-resolution Canopy Height Model (CHM) built from accurate UAV image-based point clouds [8].

Regarding poplar plots fieldwork, it was conducted in October 2022, measuring the DBH using a Pi measuring tape (millimeter precision). Tree height was measured using a Vertex IV Pro ultrasound hypsometer from the ground.

2.2 Description of UALtree Algorithm

The software UALtree, written in Matlab®, was devised as an ITD algorithm focused on detecting trees and estimate their position, height and DBH. It is based on the execution of the following stages:

(1) External preprocessing. It consisted of reducing point cloud density using a subsampling of 2 cm to streamline the subsequent stages. This process was carried out through CloudCompare 2.12.0. Next, each point cloud was automatically classified into ground and non-ground points by applying the triangular irregular network iterative approach implemented into Agisoft Metashape. The set of chosen parameters for all the reference plots were cell size = 10 m, distance = 0.3 m, and angle = 30°.

(2) Normalization, ground points removing, and height cutoff threshold. Point cloud normalization removes the influence of terrain fluctuations on the height of trees, taking the ground (digital terrain model) as height reference. UALtree builds the reference digital terrain model using the nearest neighbor algorithm. After that, the points previously labeled as ground are removed from the original point cloud. To speed up the subsequent processing, those points above a height cutoff threshold were filtered out before the segmentation step. This height threshold took a value given by the minimum tree height to be considered in the inventory plus 0.2 m. They were set to 2.2 m and 4.2 m for Aleppo pine and poplar plots, respectively.

(3) Initial tree segmentation. An initial tree segmentation was approached using the cut-pursuit 3D graph algorithm called *Treeiso*. It was devised to isolate individual tree points from plot-level TLS and PTL point clouds [9]. *Treeiso* is based on considering trees as 3D geometries with large horizontal gaps between neighboring stems, but considerably smaller 3D gaps within the tree unit. Therefore, it implements a two-stage segmentation scheme to capture the respective 3D and 2D scales. In this way, the tree point cloud is first isolated into small clusters using cut-pursuit 3D graph tunable parameters such as $K1 = 3$ (number of nearest points to search) and $\lambda 1 = 0.1$ (a regularization parameter that the larger, the more edge cuts are performed). The result are small patches of branches and crowns to be further aggregated at the individual-tree level. In the second-stage segmentation, each small patch centroid is extracted and the XY coordinates of all centroids are input to a new iteration of the cut-pursuit algorithm, in this case using $K2 = 10$ and $\lambda 2 = 5$ (same meaning as *K1* and *λ1).* Note that in this second step, the algorithm is grouping clusters more than points, so *K2* would be the number of nearest clusters to search. *K1*, *λ1*, *K2* and *λ2* were maintained constant in all the reference plots.

(4) Refinement of the initial tree segmentation. The initial tree segmentation was refined by applying a decision rule to filter out those potential tree clusters with maximum height lower than the user-defined minimum tree height (just to remove shrubs and

low vegetation) OR minimum height above 1.3 m (no points available to compute DBH). Clusters not defining an actual tree were also removed based on searching for vertical tree continuity by using the M-estimator SAmple Consensus (MSAC) algorithm to find cylindrical shapes [10].

(5) Estimates of DBH and tree position. UALtree uses a user-defined slice (0.15 m height by default) to select those points located 7.5 cm above and 7.5 cm below 1.3 m height (DBH measurement height). These selected points are projected onto the horizontal plane (XY coordinates) to robustly fit into a circle by applying the algorithm proposed by Ladrón de Guevara et al. [11].

(6) Refinement of low branches that generate confusion with stem at 1.3 m height. The cylindrically approximated DBH (see step 4) is compared to the maximum DBH expected in the stand. If the estimated DBH is higher than the expected maximum, it probably means that a low branch is contributing erroneous points to the DBH calculation slice. To solve this problem, a raster point density heat maps was generated to identify circular shapes by applying the Hough transform.

(7) Tree height estimation. It was faced by finding the maximum height of those points belonging to a cylinder with a predetermined radius (1 m by default) whose center is the previously estimated position of the tree.

Note that UALtree is only based on three real tuning parameters: the minimum tree height (to set the height cutoff threshold), the maximum DBH expected in the stand, and an experimental coefficient (K_c). K_c relates the actual and estimated DBH ($\text{DBH}_{\text{actual}} = K_c \times \text{DBH}_{\text{estimated}}$) to correct potential bias. It was observed that the value of K_c depends on the scanning device. In this work, K_c took values of 1.1 and 1 for *Faro*™ Focus3D X-330 and *GeoSLAM*™ ZEB Horizon, respectively.

2.3 Brief Description of AID-FOREST and 3D FOREST Methods

AID-FOREST is a brand-new ITD commercial software that makes use of the cross sections of stems to define circumferences that can be detected by properly trained artificial intelligence systems (Deep Learning Convolutional Neural Networks). In the case of AID-FOREST, only the sub-sampling preprocessing step (2 cm distance) was performed externally. The most important tuning parameters of AID-FOREST are the following: (i) the minimum "number of matches" or consecutive number of detected sections to determine that an object is a true tree, (ii) the "Z resolution" for detection (thickness of the slice to be rasterized), (iii) the list of heights to process (from the lowest to the highest stem height stepping Z resolution), and (iv) the "offset detection setting". This last parameter usually takes values between −0.05 and −0.09 m, which correspond to the dispersion or noise of the point cloud in contact with the stem. Actually, we only changed the "number of matches" parameter, which varied from three to five depending on the reference plot. Note that the final tree detection accuracy is sensitive to the value assigned to this parameter. In all cases, the Z resolution was set to 0.15 m, while the offset detection setting took a value of −0.05 m. The rest of tuning parameters were left as default. The reader can find a complete description about AID-FOREST in López Serrano et al. [5].

In the other corner, 3D FOREST is a widely known ITS open-source software that performs tree segmentation based on growing point clusters defined by several geometrical and topological rules. 3D FOREST version 0.5 was used in this work. It conducts the automatic tree segmentation of the vegetation cloud based on searching neighboring voxels according to the chosen feature descriptor (Principal Component Analysis (PCA) in our case). The reader can find a detailed description of the algorithm in the 3D FOREST Wiki User's Guide [12]. There are up to five tuning parameters that need to be set in 3D FOREST (in addition to the type of feature descriptor). However, we focused only on those that tend to affect the accuracy of the segmentation results the most, such as input distance (voxel size in cm) and multiplication value (a multiple of the input distance to search for another voxel when adding new free voxels to an element). After several trials with different tuning parameters, the threshold of the PCA descriptor was set at 60% of the PCA computed range, while the input distance and the multiplication value took values of 6 cm and 10, respectively. The preprocessing steps applied before running 3D FOREST were the same as those performed in UALtree.

2.4 Accuracy Assessment: Tree Detection and Features Extraction

Assessment of tree detection accuracy was addressed by matching the trees in the field data (ground truth) to the trees extracted from each method tested. A Matlab® code was written to find candidate field trees for each extracted tree (i.e., the closest field tree within a 3 m search radius). Based on the matching results for each plot, recall (r), precision (p) and F1-score (F1) metrics were computed as follows:

$$\mathrm{r} = \frac{\mathrm{TP}}{\mathrm{TP{+}FN}}; p = \frac{\mathrm{TP}}{\mathrm{TP{+}FP}}; F1 = \frac{2 \times r \times p}{r + p} \tag{1}$$

where a true positive (TP) is a field tree that matches an extracted tree, a false negative (FN) means that a field tree is not matched (omission error), and a false positive (FP) is accounted if an extracted tree does not have any matched tree from the field data (commission error). Note that r measures the under-segmentation errors, p is focused on the over-segmentation errors, while F1-score considers the overall performance by accounting both under- and over-segmentation errors.

The accuracy in estimating tree height and DBH features was also evaluated by applying some error statistics to each pair of observed and estimated values at tree level. Since the distribution of the residuals (e_i) for both DBH and H (e_i = estimated value$_i$ – observed value$_i$) usually followed a non-normal distribution (Shapiro-Wilk test), robust accuracy metrics such as median (systematic error) and median absolute deviation (MAD) (random error) were adopted. In addition, and due to its wide use, the root mean square error (RMSE) was also calculated.

3 Results and Discussion

3.1 Accuracy of Tree Detection

Figure 2 depicts the F1-score values for each reference plot and the corresponding average figures for each tested method. The average F1-score value for AID-FOREST, UALtree and 3D FOREST took values of 0.9339, 0.9378 and 0.7949, respectively, showing that both AID-FOREST and UALtree performed tree detection similarly and much better than 3D FOREST. It is interesting to note that UALtree performed better than AID-FOREST in the three poplar plots (homogeneous plots in terms of the PH^{3}50 index).

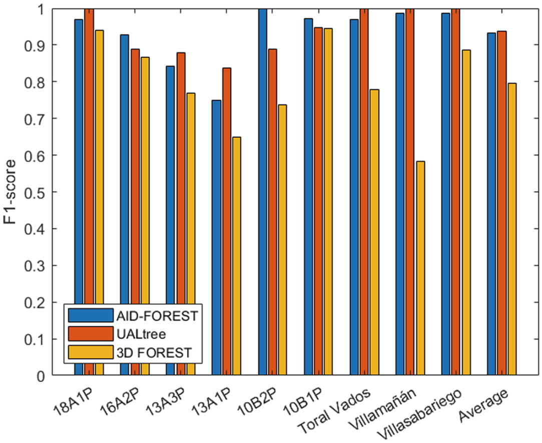

Fig. 2. Tree detection assessment for AID-FOREST, UALtree and 3D FOREST in terms of F1-score. The last cluster of bars correspond to average values.

Regarding precision and recall rates (data not shown), AID-FOREST had slightly better precision scores than UALtree (0.9539 vs. 0.9359 on average), and much better than 3D-FOREST (0.7964). This meant that UALtree made a slightly higher commission error (over-segmentation) than AID-FOREST, although also achieving an excellent score. However, and now looking at recall rates, UALtree performed slightly better than AID-FOREST (0.9441 vs. 0.9204 on average), correctly extracting more than 94% of field trees on average (omission error less than 6%). Once again, 3D FOREST had the worst recall scores (0.7416).

3.2 Accuracy of Tree Height Estimation

Figure 3 shows the box-and-whisker plots for the accuracy of the estimate of tree height (m) according to four statistics. They were computed in each plot from the differences between estimated heights minus observed height for each matched tree. In each box, the center mark indicates the corresponding median value, and the lower and upper ends of the box indicate the 25th and 75th percentiles, respectively. Whiskers extend to the most extreme data points not considered outliers. Outliers are represented by the '+' marker symbol in red. The relative values of RMSE and MAD are also depicted in Fig. 3 to intuitively scale the magnitude of the random error (in percentage) with respect to the mean value of the observed variable.

Regarding tree height accuracy estimation, both AID-FOREST and UALtree returned similar figures, noting that the median values of relative MAD were less than

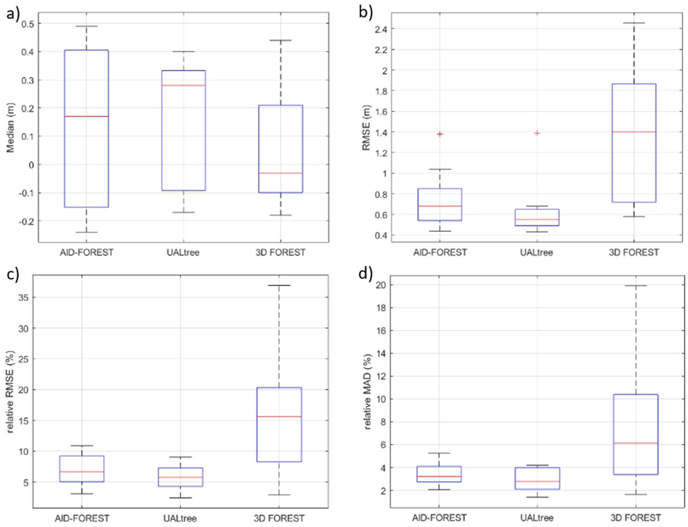

Fig. 3. Box-and-whisker plots for the accuracy of the estimate of tree height (m) according to several statistics [a) Median, b) RMSE, c) relative RMSE, and d) relative MAD].

4% in both cases. On the contrary, 3D FOREST obtained a median value of relative MAD of 6.15%, also presenting more dispersed values than AID-FOREST or UALtree. RMSE had reasonably low values of 0.68 m and 0.55 m (median values) for AID-FOREST and UAltree, respectively, while 3D FOREST had a worse performance, taking a median value of 1.4 m due to errors in tree segmentation (both over- and under-segmentation). Neither AID-FOREST nor UALtree presented significant bias in estimating tree height.

The Wilcoxon non-parametric test for paired data was applied to test whether the automatically extracted features and the field data observations were drawn from the same distribution at significance level $p < 0.05$. Table 2 reports the corresponding p-value of this test for each plot, including only the results of AID-FOREST and UALtree. The Wilcoxon test revealed that AID-FOREST was slightly more accurate than UALtree in estimating the observed tree height distribution, finding non-significant differences in six of the nine plots. UALtree produced non-significant differences in four plots, two of them being poplar plots.

Table 2. Wilcoxon (W-test) applied to tree height (H) extracted using AID-FOREST and UALtree.

Method applied	W-test p-value								
	18A1P	16A2P	13A3P	13A1P	10B2P	10B1P	Toral Vados	Villamañán	Villasabariego
AID-FOREST	0.083	<0.01*	0.255	<0.01*	0.779	0.371	0.075	<0.05*	0.069
UALtree	<0.01*	<0.01*	<0.01*	<0.01*	0.575	0.077	<0.05*	0.092	0.205

(*) Significant differences ($p < 0.05$) between estimated and observed paired values at tree level

3.3 Accuracy of DBH Estimation

Figure 4 depicts the box-and-whisker plots for the accuracy of the estimate of DBH (cm) according to median, RMSE, and relative RMSE and MAD. They were computed for each of the nine plots from the differences between estimated DBH minus observed DBH for each matched tree.

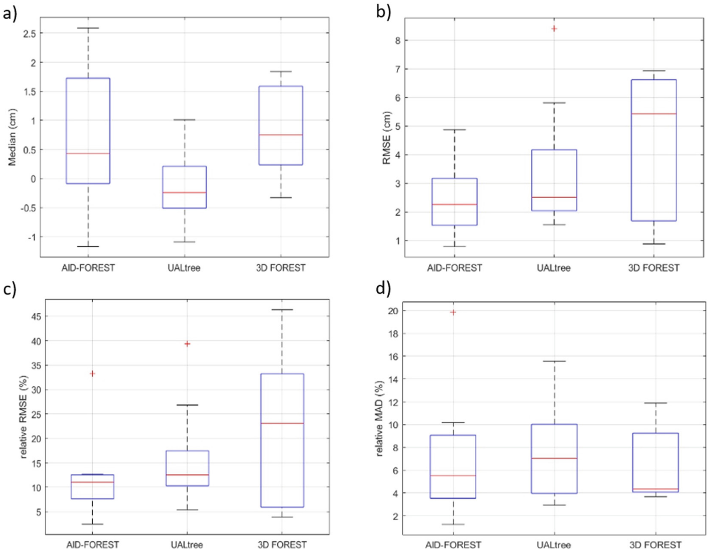

Fig. 4. Box-and-whisker plots for the accuracy of the estimate of DBH (cm) according to several statistics [a) Median, b) RMSE, c) relative RMSE, and d) relative MAD].

AID FOREST slightly outperformed UALtree in terms of random error. In fact, UALtree recorded a relative MAD of 7.03% (median value for all plots), while AID-FOREST and 3D FOREST obtained relative MAD values of 5.52% and 4.34%, respectively. It is worth considering that 3D FOREST incurred in a higher omission error, which meant that it correctly detected less trees than UALtree or AID-FOREST. On the other hand, and in terms of relative RMSE (i.e., without smoothing the presence of outliers), both AID-FOREST and UALtree presented better median values than 3D FOREST (11.03% and 12.52%, respectively, vs. 23.11%), also 3D FOREST showing a larger dispersion of relative MAD along the different plots. Summing up, 3D FOREST produced more abnormal DBH estimates. It was likely due to segmentation errors related to the presence of low vegetation and shrubs in the Aleppo pine plots.

UALtree also showed less bias in the DBH estimation than AID-FOREST or 3D FOREST, the latter tending to overestimate the observed DBH values.

Table 3 shows the p-value of Wilcoxon test for each plot. It allowed us to detect that UALtree was able to estimate the observed DBH distribution at tree level with greater precision than AID-FOREST, indicating non-significant differences between observed and estimated DBH values in up to six of the nine reference plots. Note that AID-FOREST only detected non-significant differences in three Aleppo pine plots, showing significant disparities in all poplar plots.

Table 3. Wilcoxon (W-test) applied to DBH extracted using AID-FOREST and UALtree.

Method applied	W-test p-value								
	18A1P	16A2P	13A3P	13A1P	10B2P	10B1P	Toral Vados	Villamañán	Villasabariego
AID-FOREST	0.389	0.533	0.062	<0.01*	<0.05*	<0.01*	<0.01*	<0.01*	<0.01*
UALtree	0.063	0.977	<0.05*	0.285	0.779	0.285	0.126	<0.05*	<0.01*

(*) Significant differences ($p < 0.05$) between estimated and observed paired values at tree level

3.4 Processing Time

As can be seen in Table 4, UALtree performed up to three times faster than AID-FOREST, and up to more than nine times faster than 3D FOREST. This processing time does not include external preprocessing. The key point of this improved efficiency of UALtree is the removal, before the segmentation step, of those points in the input point cloud above the height cutoff threshold.

Table 4. Processing time expressed in seconds per million points processed. All calculations were executed with Intel® CoreTM i7-8565U 4 X 1.99 GHz, 16 GB RAM, and NVIDIA Quadro P520.

Method	18A1P	16A2P	13A3P	13A1P	10B2P	10B1P	Toral Vados	Villamañán	Villasabariego	**Average**
AID-FOREST	32.69	85.08	93.50	119.73	94.58	46.80	24.93	52.81	75.68	**69.53**
UALtree	12.07	18.96	29.09	39.68	28.82	23.96	17.67	12.11	19.75	**22.46**
3D FOREST	156.29	135.50	134.30	132.38	119.34	179.82	370.32	301.10	397.54	**214.07**

4 Conclusions

UALtree proved to perform similarly to AID-FOREST in terms of F1 score, incurring slightly higher commission error but also slightly lower omission (higher recall rate). Both algorithms yielded much better accuracy rates than 3D FOREST. Furthermore, UALtree worked three times faster than AID-FOREST, and more than nine times faster than 3D FOREST.

In the case of tree height estimation, both AID-FOREST and UALtree provided similar figures, with median values of the relative MAD lower than 4% in both cases. On the contrary, 3D FOREST obtained a median value of relative MAD of 6.15%, also presenting more dispersed values. The Wilcoxon non-parametric test for paired data showed that AID-FOREST was slightly more accurate than UALtree in estimating the observed tree height distribution.

With respect to the DBH estimation, AID-FOREST and 3D FOREST slightly outperformed UALtree in terms of random error (relative MAD). However, and in terms of relative RMSE, both AID-FOREST and UALtree presented better median values than 3D FOREST (11.03% and 12.52%, respectively, vs. 23.11%). Finally, the Wilcoxon test evidenced that UALtree was able to estimate the observed DBH distribution at tree level with greater accuracy than AID-FOREST.

Acknowledgments. (1) Proyecto Retos Junta de Andalucía, Spain (P18-RT-2327). (2) Programa Operativo FEDER-Andalucía 2014–2020, Spain (UAL2020-SEJ-D1931). (3) Proyectos estratégicos orientados a la transición ecológica y digital 2021, Ministerio de Ciencia e Innovación, Spain (TED2021-132332B-C21). Thanks are also due to DIELMO 3D, which kindly provided temporary access to the AID-FOREST software. This work takes part of the Campus ceiA3, Spain.

References

1. Gómez C et al (2019) Remote sensing for the Spanish forests in the 21stcentury: a review of advances, needs, and opportunities. For Syst 28:2171–9292. https://doi.org/10.5424/fs/2019281-14221
2. Newnham GJ et al (2015) Terrestrial laser scanning for plot-scale forest measurement. Curr For Rep 1:239–251. https://doi.org/10.1007/s40725-015-0025-5
3. Cabo C, Ordóñez C, López-Sánchez CA, Armesto J (2018) Automatic dendrometry: tree detection, tree height and diameter estimation using terrestrial laser scanning. Int J Appl Earth Obs Geoinf 69:164–174. https://doi.org/10.1016/J.JAG.2018.01.011
4. Xu X, Iuricich F, De Floriani L (2023) A topology-based approach to individual tree segmentation from airborne LiDAR data. Geoinformatica 1–30. https://doi.org/10.1007/S10707-023-00487-4/METRICS
5. López Serrano FR et al (2022) Artificial intelligence-based software (AID-FOREST) for tree detection: a new framework for fast and accurate forest inventorying using LiDAR point clouds. Int J Appl Earth Obs Geoinf 113:103014. https://doi.org/10.1016/J.JAG.2022.103014
6. Trochta J, Krůček M, Vrška T, Král K (2017) 3D Forest: an application for descriptions of three-dimensional forest structures using terrestrial LiDAR. PLoS ONE 12:e0176871. https://doi.org/10.1371/journal.pone.0176871
7. Nemmaoui A, Aguilar FJ, Aguilar MA (2023) UAV-based digital terrain model generation to support accurate inventories in mediterranean forests. In: Cavas-Martínez F, Marín Granados MD, Mirálbes Buil R, De-Cózar-Macías OD (eds) Advances in design engineering III. INGEGRAF 2022. lecture notes in mechanical engineering. Springer International Publishing, Cham, pp 565–581
8. Aguilar FJ, Nemmaoui A, Aguilar MA, Jiménez-Lao R (2022) Aleppo pine allometric modeling through integrating UAV image-based point clouds and ground-based data. ISPRS Ann Photogramm Remote Sens Spat Inf Sci V-3–2022, pp 353–360. https://doi.org/10.5194/ISPRS-ANNALS-V-3-2022-353-2022
9. Xi Z, Hopkinson C (2022) 3D graph-based individual-tree isolation (Treeiso) from terrestrial laser scanning point clouds. Remote Sens 14:6116. https://doi.org/10.3390/RS14236116
10. Torr PHS, Zisserman A (2000) MLESAC: a new robust estimator with application to estimating image geometry. Comput Vis Image Underst 78:138–156. https://doi.org/10.1006/CVIU.1999.0832

11. Ladrón de Guevara I, Muñoz J, de Cózar OD, Blázquez EB (2011) Robust fitting of circle arcs. J Math Imaging Vis 40:147–161.https://doi.org/10.1007/s10851-010-0249-8
12. Home VUKOZ-OEL/3d-forest-classic Wiki GitHub. https://github.com/VUKOZ-OEL/3d-forest-classic/wiki. Accessed 05 Apr 2023