Automatic Individual Tree Detection from Combination of Aerial Imagery, LiDAR and Environment Context

Daniel Amigo(✉), David Sánchez Pedroche, Jesús García, and José M. Molina

Group GIAA, University Carlos III of Madrid, Madrid, Spain
{damigo,davsanch,jgherrer,molina}@inf.uc3m.es

Abstract. Geographic Information Systems (GIS) allow analysis based on geo-referenced data. Currently only simple geo-referenced information is available, such as road networks or types of terrain, but there are other geo-referenced data that would be very useful to facilitate decision-making. These data are not collected as they are very hard to generate manually, but remote sensing data and artificial intelligence can be used to accomplish it. This work aims to develop an automatic framework for the extraction of geo-referenced trees, through the union Light Detection and Ranging (LiDAR) point clouds, aerial imagery, and existing GIS environment context. The results of the process are satisfactory, improving in some several areas the LiDAR-based detections using only imagery. However, issues such as false positives need to be corrected in the future. Merging both data sources would allow better results to be achieved.

AQ1

Keywords: Individual tree location · GIS data · LiDAR processing · Aerial imagery · Tree counting · Airborne LiDAR data

1 Introduction

Georeferenced information has been essential for decision-making for centuries. Cartography traditionally produced manual and reliable representations of the surface of the Earth, as elevation contours, terrain types, or road networks. As computers grew, this information moved to a digital format. Today, advances in sensing techniques make possible to obtain more reliable information through automated processes, resulting in more detailed digital maps.

Geographic Information Systems (GIS) are software applications capable of processing maps and geo-referenced data, allowing the analysis of all kinds of information, such as images of the earth's surface, polygons representing types of urban areas or objects, or vectors forming road networks. There is open-source geo-referenced data, such as terrain elevation models or road networks. However, specific datasets that could be useful for research have not yet been extracted, stored, or published online.

The identification of individual objects is, in some cases, a very time-consuming task to perform manually. However, thanks to new algorithms and sensors combined

H. Sanjurjo González et al. (Eds.): SOCO 2021, AISC 1401, pp. 1–10, 2022.
https://doi.org/10.1007/978-3-030-87869-6_28

with improved computing capabilities, it is possible to perform this task automatically and with greater accuracy. Nevertheless, current sensors still do not fully capture the environment, so it is common to fuse several sensors' information to improve positioning and accuracy. Trees are an important item not available individually in GIS datasets. Climate change makes them a critical element for the future and having a database which identifies each tree uniquely could benefit scientific analysis. It could also help with other tasks, such as forest risk and fire mitigation, or the rapid recognition and response to other real problems, such as a fungus drying out holm oaks in Spain [1, 2].

This paper proposes a solution that combines two low-resolution sensors to extract trees' positions automatically. It uses the LiDAR sensor to generate detections based on the elevation of the point cloud automatically. These detections are transformed to bounding boxes in an orthophoto, allowing a neural network to learn the pixels that form a tree, thus automatically detecting other trees that are not detected by the LiDAR due to its low resolution. In addition, during the process, existing GIS information is exploited, such as roads or buildings, where trees cannot be found, solving some of the drawbacks of LiDAR approaches. The results of the process are promising, achieving results like those obtained by LiDAR, but with better adjustment of the bounding-box on the tree. Even so, some problem cases should be solved in the future. A fusion of all data in the same procedure would allow a more accurate mapping of the trees.

This paper is organized as follows: Sect. 2 explains the sensors data used in this paper and state-of-the-art methods for detecting individual trees with them. Section 3 introduces the proposed framework in detail, and Sect. 4 evaluates the results through illustrative examples of the framework compared to using LiDAR alone. Finally, the conclusions and perspectives for future works are presented in Sect. 5.

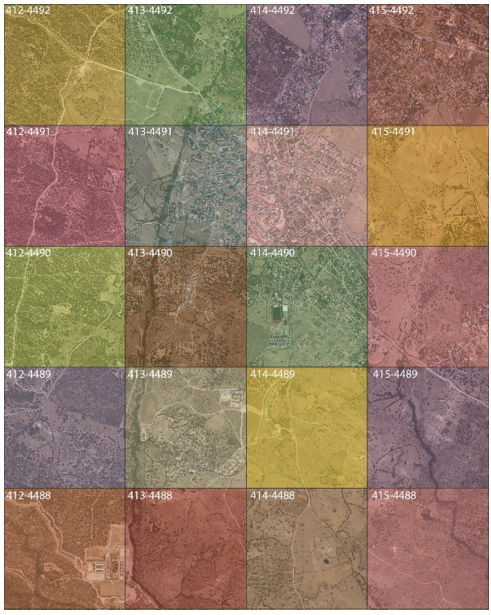

Fig. 1. Orthophoto crop and LiDAR quadrants

2 Related Work

The detection of individual trees using machine learning and deep learning techniques is not a new research field, but it has regained interest in the last decade, as the technology allows the problem to be handled faster and more accurately. Researchers have approached the problem in different ways, depending on the data used to extract detections. This section explores several approaches for the sensors used in this work.

The dataset used is provided by the Spanish National Aerial Orthophotography Plan (PNOA) [3]. Carried out in 2016 through aerial photogrammetric flights, provides information of high-precision cameras and multispectral cameras with a spatial resolution of 25 cm and LiDAR point clouds with different densities across the territory. For this paper's experiments, a subset of Colmenarejo is used, the town where one of the campuses of the University Carlos III of Madrid (UC3M) is. Figure 1 shows the orthophoto fragment and the 20 LiDAR sets to be used. Specifically, it is an extension of 20 km^2, where a total of 51.5 million LiDAR points were captured. This generates an average of 2.575 points per m^2, a low number to perform an accurate detection. Because of this low resolution, solutions that combine different sensors are also being explored.

2.1 LiDAR-Based Tree Detection Approaches

A LiDAR device, which stands for Light Detection and Ranging, measures the distances from the device to multiple points around it by calculating the delay between an emission and its subsequent detection. These points, knowing the device's position, can be geopositioned, thus generating a three-dimensional space that corresponds to reality. LiDAR technology is becoming more and more popular due to its low cost and capabilities, for instance in unmanned aerial vehicle (UAV) flights.

Using it from aerial view pointing to the surface, these points can be used for tree detection by measuring the variation in altitude of nearby points. However, object detection requires high sensor resolution or multiple flights at low altitude to achieve good results. If the point density is low, it may not detect trees or even generate false positives, e.g., with poles or lampposts. Li et al. [4], Hamraz et al. [5], Jeronimo et al. [6], Liu et al. [7], Silva et al. [8] are some of the most relevant works in identifying individual trees from LiDAR.

LiDAR can also be used for object detection from other perspectives. Babahajiani et al. [9] detects trees using 3D LiDAR at street level, but its structure is too complex for this use.

2.2 Aerial Imagery-Based Tree Detection Approaches

A second approach is to use aerial imagery. Satellites, aircraft flights or unmanned aerial vehicles can take high-resolution snapshots, on a fairly regular basis and periodically. Using the RGB channels of these images, deep learning can be performed to detect trees.

Another approach to the problem is to use aerial imagery. Satellites, airborne flights, or UAVs can take high-resolution snapshots, in a very regular and periodic manner. Deep learning can be performed using their RGB channels to detect trees.

Schnell [10] reviews tree monitoring methods, where many image -based and satellite-based works are highlighted. Malkoç [11] detects trees outside forests from satellite elevation, but does not perform an individual tree segmentation. Ardila et al. [12] uses satellite special sensors: pan-chromatic and multispectral layers.

As with LiDAR, static terrestrial objects can be geolocated using images from the Earth's surface. The approaches are different, taking advantage of the detection of the same object from different perspectives, as Lumnitz et al. [13] or Laumer et al. [14] does.

2.3 Sensor Fusing Tree Detection Approaches

Several researchers have identified the need to fuse several sensors in one process to obtain better results in several ways: avoiding false positives, adding additional information, or improving accuracy.

Silva et al. [8] uses aerial LiDAR detection, but uses it to complement and add information on GPS detections next to each tree, merging both detections and correcting possible noise from GPS measurements. Wegner and Branson [15, 16] fuses detections by means of imagery, mainly street view, but also accepts aerial. Each detection is calculated separately, and they are fused by giving at each measure a specific weighting.

3 Proposed Framework

The aim of this work is to develop a process that takes advantage of two complementary sensors with low resolution, LiDAR, and aerial imagery, to obtain a better result in the tree detection. This section presents the description of the algorithms used and their workflow. Figure 2 illustrates this process structure, composed of three main blocks:

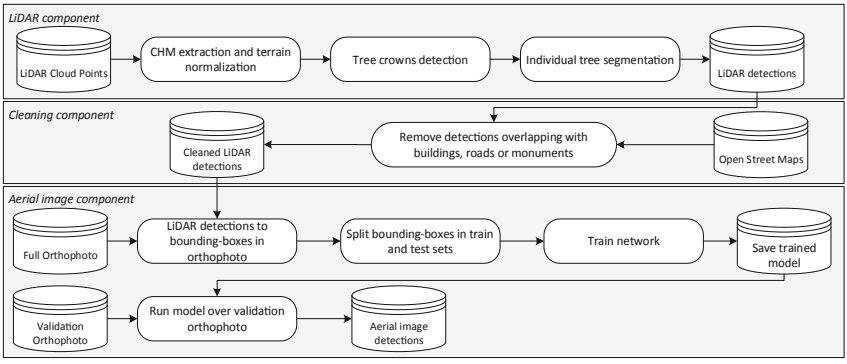

Fig. 2. Tree detection process

1. A LiDAR component extracts individual trees from the input point cloud.
2. A cleaning process removes erroneous detections using context information.
3. The individual tree detection component using imagery. Using the prior cleaned detections, they are converted to the image and used to training a prediction model.

3.1 LiDAR Component

The LiDAR component is developed using the LidR library [17]. This R package provides tools for tree detection with LiDAR data. First, the points on the ground surface, which could not be trees, are removed from the point cloud, easing further processes. To do this, the Digital Terrain Model (DTM), the terrain level, is extracted and normalized, eliminating the variation in ground level, setting the entire area to the same elevation. With this normalised terrain, the Canopy Height Model (CHM) is extracted, containing only the points elevated above the ground.

Next, the Local Maximum Filter (LMF) algorithm [8] is used on the CHM to detect tree crowns. By sliding a window over the point cloud, it finds the local maximum points in height, each one being a tree at the output of the process. This algorithm is essential in the LiDAR component, and therefore requires that the window size is appropriate for the input data, so that it does not miss when several trees are nearby, but only detects the highest one. Specifically, in this work, local maximums are found from a minimal AQ2 height of 1.5 m, and a circular window of 3.5 m in diameter is used (Fig. 3).

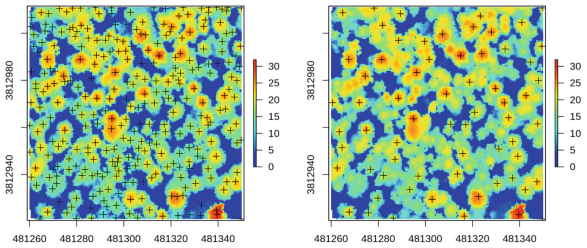

Fig. 3. Local maxima filter example (+ marker). Blue represent DTM, the rest is CHM [17]

Once the canopy positions are known, the segmentation process is performed by defining which points in the point cloud belong to each tree. For this process, the algorithm of Silva et al. [8] has been selected, as it obtained good results with the default parameters [18]. If the detailed height and number of trees were known, a manual adjustment would have to be applied, but in the problem this is unknown. This segmentation step uses the Voronoi tessellation with the detected crowns as centroids, calculating the area of each individual tree over the remaining CHM points.

3.2 Cleaning Component Using GIS Data Context

The LMF process produces well-known false positives, as it only picks through the point elevation, failing at lampposts, road signs, buildings, or any other object. As LiDAR segmentation algorithms do not have any protection and generate erroneous trees, this process has been developed, proposing a post-cleaning algorithm to eliminate such erroneous detections, whenever they are known to be wrong.

For this, the context of the area provided by OpenStreetMaps (OSM) is used, eliminating false tree detections in land areas that are unlikely to contain a tree, such as buildings, roads, sports facilities, or monuments. Figure 4 illustrates its functioning on the university campus. Note that it is important that the GIS data used must be well matched. Trees can be very close to roads or buildings, so a criterion for removal is added, only if the tree overlaps more than 50% of its area with the desired surface.

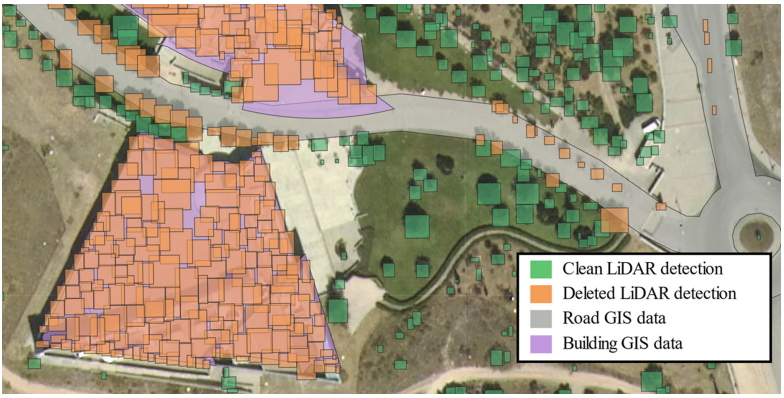

Fig. 4. Context cleaning on LiDAR detections

3.3 Imagery Component

After cleaning the LiDAR detections, the image detection component is next. But first, it is necessary to transform the LiDAR detections to bounding boxes in the image pixels. Figure 5 shows this transformation process, converting each bounding-box from the global coordinate system (WGS-84) to the image coordinate system (EPSG:3345).

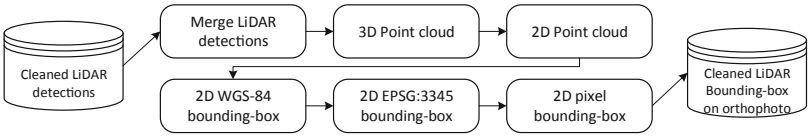

Fig. 5. LiDAR detections to orthophoto

Once the detections in the image are determined, they are divided into training, test, and validation. For validation, the quadrant 414–4489 containing the university campus

(see Fig. 1) is chosen, while the rest is divided into 70% training and 30% testing randomly. DeepForest [19, 20] is the deep learning software used to detected the trees. It is based on RetinaNet, a good and proven solution for aerial image detection [21].

After the whole process, a trained model capable of detecting trees only by aerial imaging is obtained. For it to work properly, the prediction image should have a similar terrain type, trees, and resolution to those used for training.

This model adds robustness to the process, finding trees that the LiDAR component might have missed due to its low resolution, and even recovering trees that were incorrectly removed in the cleaning process.

4 Results Analysis

After applying the LiDAR detection component in the 19 quadrants for training and testing, 263,509 trees are detected. Of these, 20,044 are removed because they coincide with buildings or sports fields. Another 1,406 are removed because they overlap with roads (although many of them are not mapped in OSM). The remaining trees are used to train the image of quadrant 414–4489 in Fig. 1.

As there is no individual tree dataset for the area, the analysis performed is based on several images (Fig. 7, Fig. 8 and Fig. 9) to comment several remarkable findings, and by comparing the LiDAR detections along the quadrant and those generated by the imagery model explained above. Specifically, 4,759 are the tree detections in the university quadrant using only LiDAR and cleaning, while 3,816 are with the full process.

There are cases where the model generates false positives, especially in urban areas, due to clear colour variations caused by the presence of tree-like shadows. It is also difficult to detect trees within the pavement, probably due to the limited training examples, as the campus is a different environment.

However, the LiDAR component can detect trees with a small area, typical in the campus, which the vision model is unable to detect because they are so narrow. Conversely, it can also be seen how the network detects large trees slightly better, with one bounding-box where LiDAR tends to generate multiple ones or doing it too small for the tree area. This behaviour is due to the limitations of the LMF algorithm, as some trees are not detected and, in some cases because of the low resolution of the LiDAR, they are not correctly modelled in the segmentation process. Figure 6 illustrates this bounding box size difference between the two series.

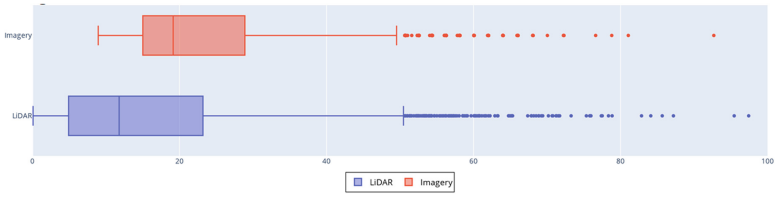

Fig. 6. Area of bounding-boxes in LiDAR and imagery (squared metres)

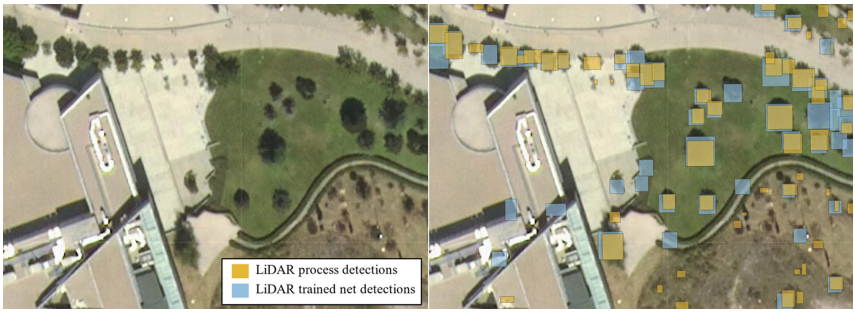

Fig. 7. First results comparison

Figure 9 shows how the LiDAR process errs with the university monument generating fake detections. As such geolocated information is not available, these detections have not been eliminated in the cleaning process, thus the imagery model has learned this pattern. In addition, due to the window used in the LMF algorithm, it does not generate detections of nearby trees either, being the robustness of the image system the one that detects them. Nevertheless, Fig. 8 shows a non-urban area with high density of trees and the same behaviour can be observed: larger bounding-boxes and some additional ones by the full process are found with respect to the LiDAR and cleaning processes.

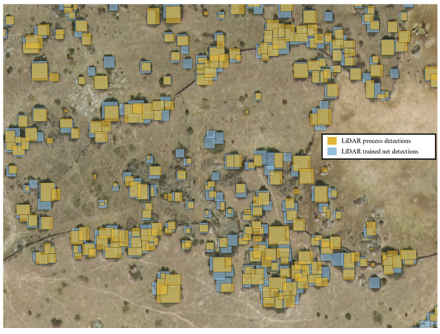

Fig. 8. Non-urban results

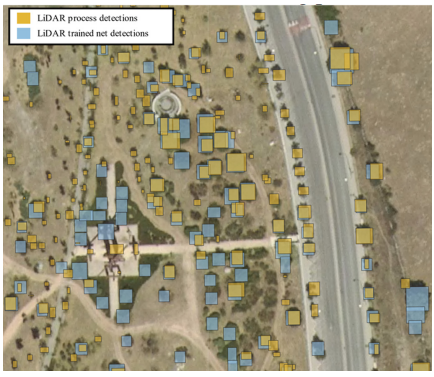

Fig. 9. Results near monument and small trees

5 Conclusions and Perspectives

This work demonstrates the potential of automatically generating GIS datasets by using only aerial imagery. Also, the transfer of LiDAR detections to imagery detections for training allows the generation of an automatic process with a lot of potential. The results are positive, but this first version needs to be refined before real use, solving several existing problems due to the lack of sensitivity of LiDAR or the false positives encountered by the imagery model. The integration of existing GIS data to improve detection solutions is an unexplored branch of the literature which shows potential.

Further research can be developed in a variety of directions:

- Fine-tune of the current process, adjusting its current algorithms or designing new ones, by integrating both information sources into a unified process, aiming for better detection results.
- Validate the process with different scenarios, ideally with high quality data. One option is to generate an own dataset by flying UAVs with sensors equipped at a lower altitudes and different angles can also be incorporated. Trying to extract additional information from the GIS layers, such as tree type, facilitating real analysis, which requires a greater amount of information that distinguishes the tree.
- Exploiting and adapt the process developed for the detection of other useful static information, such as building walls and fences, or power lines.
- Finally, the application of this generated GIS data to demonstrate its utility. For example, a current open line is the realistic recreation of the three-dimensional environment to perform UAV flight simulations, in which tree location is very useful.

Acknowledgement. This work was funded by public research projects of Spanish Ministry of Economy and Competitivity (MINECO), reference TEC2017-88048-C2-2-R.

References

1. Vivas, M., Hernández, J., Corcobado, T., Cubera, E., Solla, A.: Transgenerational induction of resistance to Phytophthora cinnamomi in Holm Oak. Forests **12**, 100 (2021). https://doi.org/10.3390/f12010100
2. Rodríguez-Romero, M., Godoy-Cancho, B., Calha, I.M., Passarinho, J.A., Moreira, A.C.: Allelopathic effects of three herb species on Phytophthora cinnamomi, a pathogen causing severe oak decline in mediterranean wood pastures. Forests **12**, 285 (2021). https://doi.org/10.3390/f12030285
3. Instituto Geográfico Nacional, Centro Nacional de Información Geográfica: Plan Nacional de Ortofotografía Aérea. https://pnoa.ign.es/
4. Li, W., Guo, Q., Jakubowski, M.K., Kelly, M.: A new method for segmenting individual trees from the LIDAR point cloud. Photogramm. Eng. Remote Sens. **78**, 75–84 (2012). https://doi.org/10.14358/PERS.78.1.75
5. Hamraz, H., Contreras, M.A., Zhang, J.: A robust approach for tree segmentation in deciduous forests using small-footprint airborne LiDAR data. Int. J. Appl. Earth Obs. Geoinf. **52**, 532–541 (2016). https://doi.org/10.1016/j.jag.2016.07.006
6. Jeronimo, S.M.A., Kane, V.R., Churchill, D.J., McGaughey, R.J., Franklin, J.F.: Applying LiDAR individual tree detection to management of structurally diverse forest landscapes. J. Forest. **116**, 336–346 (2018). https://doi.org/10.1093/jofore/fvy023
7. Liu, J., Shen, J., Zhao, R., Xu, S.: Extraction of individual tree crowns from airborne LiDAR data in human settlements. Math. Comput. Model. **58**, 524–535 (2013). https://doi.org/10.1016/j.mcm.2011.10.071
8. Silva, C.A., et al.: Imputation of individual longleaf pine (Pinus palustris Mill.) tree attributes from field and LiDAR data. Can. J. Remote Sens. **42**, 554–573 (2016). https://doi.org/10.1080/07038992.2016.1196582

9. Babahajiani, P., Fan, L., Kämäräinen, J.-K., Gabbouj, M.: Urban 3D segmentation and modelling from street view images and LiDAR point clouds. Mach. Vis. Appl. **28**(7), 679–694 (2017). https://doi.org/10.1007/s00138-017-0845-3
10. Schnell, S., Kleinn, C., Ståhl, G.: Monitoring trees outside forests: a review. Environ. Monit. Assess **187**(9), 1–17 (2015). https://doi.org/10.1007/s10661-015-4817-7
11. Malkoç, E., Rüetschi, M., Ginzler, C., Waser, L.T.: Countrywide mapping of trees outside forests based on remote sensing data in Switzerland. Int. J. Appl. Earth Obs. Geoinf. **100**, 102336 (2021). https://doi.org/10.1016/j.jag.2021.102336
12. Ardila, J.P., Tolpekin, V.A., Bijker, W., Stein, A.: Markov-random-field-based super-resolution mapping for identification of urban trees in VHR images. ISPRS J. Photogramm. Remote Sens. **66**, 762–775 (2011). https://doi.org/10.1016/j.isprsjprs.2011.08.002
13. Lumnitz, S., Devisscher, T., Mayaud, J.R., Radic, V., Coops, N.C., Griess, V.C.: Mapping trees along urban street networks with deep learning and street-level imagery. ISPRS J. Photogramm. Remote Sens. **175**, 144–157 (2021). https://doi.org/10.1016/j.isprsjprs.2021.01.016
14. Laumer, D., Lang, N., van Doorn, N., Mac Aodha, O., Perona, P., Wegner, J.D.: Geocoding of trees from street addresses and street-level images. ISPRS J. Photogramm. Remote Sens. **162**, 125–136 (2020). https://doi.org/10.1016/j.isprsjprs.2020.02.001
15. Wegner, J.D., Branson, S., Hall, D., Schindler, K., Perona, P.: Cataloging public objects using aerial and street-level images — urban trees. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 6014–6023. IEEE (2016)
16. Branson, S., Wegner, J.D., Hall, D., Lang, N., Schindler, K., Perona, P.: From google maps to a fine-grained catalog of street trees. ISPRS J. Photogramm. Remote Sens. **135**, 13–30 (2018). https://doi.org/10.1016/j.isprsjprs.2017.11.008
17. Roussel, J.-R., et al.: lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. Remote Sens. Environ. **251**, 112061 (2020). https://doi.org/10.1016/j.rse.2020.112061
18. Zaforemska, A., Xiao, W., Gaulton, R.: Individual tree detection from UAV LIDAR data in a mixed species Woodland. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. **XLII-2/W13**, 657–663 (2019). https://doi.org/10.5194/isprs-archives-XLII-2-W13-657-2019
19. Weinstein, B.G., Marconi, S., Bohlman, S., Zare, A., White, E.: Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. Remote Sens. **11**, 1309 (2019). https://doi.org/10.3390/rs11111309
20. Weinstein, B.G., Marconi, S., Aubry-Kientz, M., Vincent, G., Senyondo, H., White, E.P.: DeepForest: a Python package for RGB deep learning tree crown delineation. Methods Ecol. Evol. **11**, 1743–1751 (2020). https://doi.org/10.1111/2041-210X.13472
21. Li, K., Wan, G., Cheng, G., Meng, L., Han, J.: Object detection in optical remote sensing images: a survey and a new benchmark. ISPRS J. Photogramm. Remote Sens. **159**, 296–307 (2020). https://doi.org/10.1016/j.isprsjprs.2019.11.023

Author Queries

Chapter 28

Query Refs.	Details Required	Author's response
AQ1	Please check and confirm if the authors given and family names have been correctly identified.	
AQ2	Please check and confirm if the inserted citation of Fig. 3 is correct. If not, please suggest an alternate citation.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	··· under matter to remain	New matter followed by
Insert in text the matter indicated in the margin		or (x2)
Delete	through single character, rule or underline or through all characters to be deleted	or (x2)
Substitute character or substitute part of one or more word(s)	through letter or through characters	new character or new characters
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	through character or where required	
Reduce space between characters or words	between characters or words affected	