TREE DIGITISATION FROM POINT CLOUDS WITH UNREAL ENGINE

S. González-Domínguez, J. Balado*, A. Novo, P. Arias

GeoTECH, CINTECX, Universidade de Vigo, 36310 Vigo, Spain
serggonzalez@alumnado.uvigo.gal, (jbalado,annovo,parias)@uvigo.gal

KEY WORDS: LiDAR, Game Engines, Terrestrial Laser Scanning, Mobile Laser Scanning, Geometric Features.

ABSTRACT:

Trees are fundamental parts of urban areas and green urbanism. Although much effort is being put into the digitisation of urban areas, trees present great complexity and are usually replaced by predefined models. On the one hand, trees are elements composed of trunk, branches, and leaves, each with a completely different structure and geometry. On the other hand, the tree parts are closely related to each species. Therefore, in order to obtain a realistic digital urban environment, in 3D models such as CityGML or Metaverse, it is necessary that the trees correspond faithfully to reality. The aim of this work is to propose a method to digitise trees from Mobile Laser Scanning and Terrestrial Laser Scanning data. The process takes advantage of the differentiation between trunks and leaves for their segmentation by point cloud geometric features. Unreal Engine is then used to digitise each part. Trunk and branches are geometrically preserved. For dense canopy trees, predefined leaves according to the species are imported and the alpha shape of the crown is filled. For non-dense canopy trees, the canopy is imported and modified to fit the branches. The method was tested on four real case studies. The results show realistic trees, with correct trunk and foliage segmentation, but highly dependent on the life/canopy repositories. Unreal Engine was a very complete and useful tool for the digitisation of trees generating realistic textures and lighting options.

1. INTRODUCTION

Virtual Reality (VR) was born during the 60s of the 20th centuries (Delaney, 2016) and is a set of computer techniques that make it possible to create simulated images and spaces. These spaces enable users to come in and out of the boundary between reality and imagination. To develop interactive content with 3D virtual space, a virtual environment should be studied to recreate reality. Currently, much of the progress in the field of virtual reality is focused on representing reality as accurately as possible and serving as a model for different activities.

LiDAR technology allows a 3D representation of reality more efficiently and quickly than other methods (Guan et al., 2016). In addition, LiDAR forestry data have great potential for analysis using VR because of their inherently 3D nature (Gardner et al., 2003). Nevertheless, for full integration of LIDAR analysis with VR, the range of analytical tools needs to be greatly increased. Since, LiDAR system can lead errors in the data collection specially of trees and dense vegetation (Simpson et al., 2017). Some of these errors occur due to the point cloud densification in the areas of branches and leaves or for the difference in the shape of trees that prevents the establishment of similarities for the image classification. As consequence, the noise filtering generated by LiDAR acquisition in the tree environment is not an easy task to perform. Therefore, the development of algorithms to improve trees treatment would lead to considerable reduce times in trees recreation. Additionally, these algorithms allow the development of improvements in both LiDAR and VR, applying systems such as Oculus and Google Leans.

The aim of this work is to develop a method to digitise trees from point clouds while preserving their geometry. The method is implemented in two well-known and widespread software: Cloud Compare (Girardeau-Montaut, 2016) for manual point cloud processing, and Unreal Engine for visualisation of virtual environments, which is a Game Engine used in VR and architectural visualization, so high-quality 3D contents are easily accessible (Sanders, 2016).

2. RELATED WORK

A comprehensive review of tree inventory using remote sensing techniques can be found in (White et al., 2016). Regarding the methods for tree modelling from point clouds, there are four main research lines that laid the base for tree modelling.

The first of which set the fixed standards, such as the Space colonization algorithm (Runions et al., 2007) which generate trees and shrubs by simulating the competition for space between growing branches. This method starts with a surface revolution obtained around the tree axis which determines the space available for growth and is subsequently eliminated by the segment creation corresponding with the different branches. Then, the tree geometry is modelled using cylinders centred on the tree skeleton axes that are generated through the diameter calculation of each branch, the distance to the central node and the altitude above the ground. Finally, the necessary leaves or flowers are added. The method applied to Terrestrial Laser Scanning (TLS) data is presented in (Raumonen et al., 2013).

The second research line uses procedural modelling by generating parametric context-free (Šťava et al., 2010) which compares a tree with models generated by defined algorithms with specific measurement parameters. These parameters can be grouped into three types: geometric, seed nature and environmental. Geometric parameters define the tree shape, for example by determining the internodal distance between two branches. The seed nature parameters define the structure and

* Corresponding author

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-555-2023 | © Author(s) 2023. CC BY 4.0 License.

density of the branch. And the environmental parameters define the growth direction due to light incidence and gravity. Finally, once defined the parameters then are compared cyclically with the initial tree model to the optimal model reached.

The third research line is based on the deep photo technique (Kopf et al., 2008) which are designed to obtain 3D models in a more intuitive way. These methods depend on a tree database which is compared with the 2D tree projections to be modelled then, for the matches obtained, branches and trunk are reconstructed. Also, the user provides some tracings of branches, or the shape of the crown and the most similar parameters are selected building the 3D model. Then, the branches are positioned according to the position of adjacent branches by the self-similarity: random selection of replication blocks, scaling, reorienting, and joining the branches. The user can limit the preparation of the branches or propagated up to a fixed number. Finally, the user can select the leaves from the dataset or use the default one associated with the modelled tree template.

Lastly, the fourth method stablish the use of images (Reche-Martinez et al., 2004; Tan et al., 2007) which is divided in three steps, first the image is captured, second the 3D point reconstruction is performed and finally, branch and leaf recovery is carried out. The method is designed to minimize user involvement as much as possible by using image data and implementing the use of photogrammetry as an alternative to differentiate parts of the tree. However, the reconstruction of branches is one of the most complex processes. For the visible branches, first a graph is constructed with the 3D points and the 2D projection information, then by means of functions based on mathematical models (Pielou, 1966), the graphs are joined and the branches are created. Another option is to refine the tree skeleton by adding or deleting nodes by the user. For non-visible branches, the visible branches by are recovered by reconstructing the visible branches by using replication blocks which uses growth patterns for occluded branches. Finally, the corresponding leaves are added, using simple parameters (such as leaves pointing away from the branches) or the original image can be traversed by segmentation and grouping the leaves to create 3D model considering the different textures to be transferred and implemented in the previously processed tree.

In short, the methods described above have laid the foundation for 3D tree modelling. However these methodologies have been optimized by generating tree branch structures from laser scanned terrestrial clouds (Cheng et al., 2007), generating methodologies capable of extracting tree structures (Xu et al., 2007), reconstructing multiple pre-segmented trees (Livny et al., 2010), or generating real-time models suitable for interactive rendering, furthermore to classify different tree species re-samples using different processing parameters (Livny et al., 2011). This work starts from the part modelling (trunk/leaves) proposed by other authors and then focuses on the evaluation of Unreal Engine to achieve a more realistic model that can be directly imported into VR software.

3. METHODOLOGY

The proposed method is based on a tree digitization by parts: trunk and foliage. Therefore, it is necessary in a first step to identify in the point cloud the trunk-branches and separate them from the leaves. Subsequently, depending on the density of the foliage (Johnson et al., 2021), the 3D volume of the canopy is filled with (dense crown trees), or a predefined crown is imported and adjusted to the branches (non-dense crown trees). The tree digital model is visualized in Unreal Engine 4 (UE4). Figure 1 shows the workflow of the proposed method.

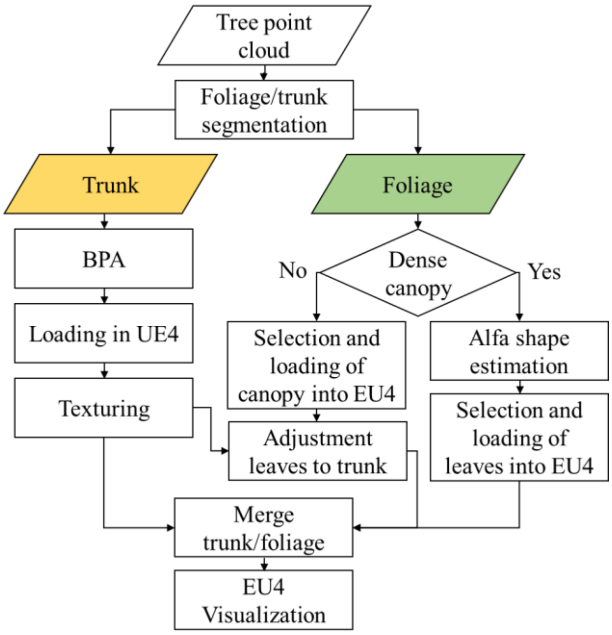

Tree point cloud
Foliage/trunk segmentation
Trunk
Foliage
BPA
Dense canopy
No
Yes
Loading in UE4
Selection and loading of canopy into EU4
Alfa shape estimation
Texturing
Adjustment leaves to trunk
Selection and loading of leaves into EU4
Merge trunk/foliage
EU4 Visualization

Figure 1. Workflow of the proposed method

3.1 Foliage/trunk segmentation

The foliage/trunk segmentation is performed from the calculation of local geometric features (Weinmann et al., 2015). For this, the eigenvalues (e_1, e_2, e_3) represent the extent of a 3D ellipsoid of each point of the input cloud $P(P_X, P_Y, P_Z)$ according to a neighbourhood of $k = 25$ neighbours. These features have been shown to be very useful in object segmentation and classification (Balado et al., 2023). The calculated features are linearity (Eq.1), planarity (Eq.2), sphericity (Eq.3), first and second principal components (Eq.4-5), curvature (Eq.6), eigentropy (Eq.7), omnivariance (Eq.8) and verticality (Eq.9). These geometric features are then analysed to identify those most useful for trunk/foliage segmentation. In addition, a Statistical Outlier Removal filter was applied to the trunk point cloud to remove isolated points.

$$linearity = \frac{e_1 - e_2}{e_1} \tag{1}$$

$$planarity = \frac{e_2 - e_3}{e_1} \tag{2}$$

$$sphericity = \frac{e_3}{e_1} \tag{3}$$

$$first\_principal\_component = \frac{e_1}{(e_1 + e_2 + e_3)} \tag{4}$$

$$second\_principal\_component = \frac{e_2}{(e_1 + e_2 + e_3)} \tag{5}$$

$$curvature = \frac{e_3}{(e_1 + e_2 + e_3)} \tag{6}$$

$$eigentropy = -\sum_{i=1}^{3} e_i \ln(e_i) \tag{7}$$

$$omnivariance = \sqrt[3]{e_1 e_2 e_3} \tag{8}$$

$$verticality = 1 - |N_Z| \tag{9}$$

3.2 Trunk digitization

The meshing of the trunk is performed by means of the Ball Pivoting Algorithm (BPA) (Bernardini et al., 1999), being previously necessary the calculation of surface normals $N(N_X, N_Y, N_Z)$. The BPA is based on rolling a ball around the cloud points while generating triangles. BPA assumes that the points are all in object surfaces. The algorithm starts from three random points on which the ball is placed. By always maintaining contact with two of these points, the ball pivots until it touches another nearby point. These three points form a triangle and so on. A radius r for the ball is needed, assuming the ball cannot cross the surface without touching the points since the radius of the ball is greater than the minimum distance between neighbourhoods. Because of this, r is taken to be twice the average distance between neighbouring points. The trunk generated mesh is loaded into Unreal Engine and assigned the most suitable wood texture from Quixel Bridge (*Quixel Bridge*, n.d.).

3.3 Dense canopy digitization

3.3.1 Canopy delimitation: For the digitization of dense canopy trees, the 3D volume that encloses the canopy is calculated by means of a triangulation and an alpha shape. The aim of using alpha shape is to preserve the shape of the tree, not to segment the canopy from the rest of the environment. The convex hull is created by means of the Delaunay triangulation. The main advantage of this triangulation is the adaptation to complex surfaces and volumes. An alpha shape is formed from the limit of an alpha complex, which is a subcomplex of the Delaunay triangulation. For this set, a family of alpha shapes can be defined, with the parameter α, that specifies the level of refinement and is adjusted experimentally (Section 4.3).

3.3.2 Leaf selection and import: The leaf is one of the most distinctive parts of the tree. In (Shutterstock, n.d.), there is an open repository of leaves (Figure 2). The model that most closely matches the tree species is selected and imported into Unreal Engine using the foliage editor as a static mesh. The foliage density (number of leaves per unit area) is adjusted depending on the tree species. Also, random rotation/tilt angles are defined, while preserving the predefined imported leaf size and texture.

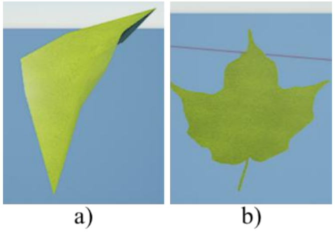
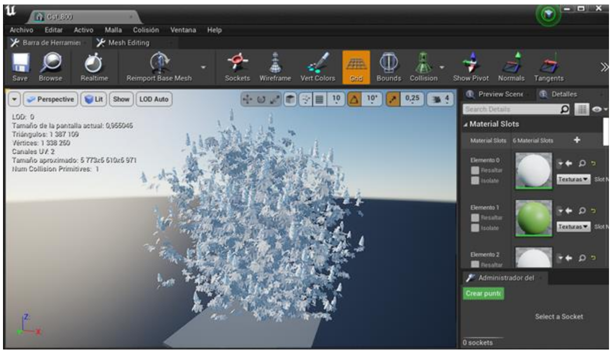


Figure 2. Examples of available free leaf models.

3.4 Non-dense canopy digitization

For the digitization of non-dense tree canopies, external canopy models available from (Shutterstock, n.d.) are used, not single leaves as in dense canopy digitization. The tree canopy that best fits the species is selected. Then, the canopy is imported into Unreal Engine and using the mesh editor (Figure 3), the foliage is manually adjusted to the branches, moving, or removing distant leaves.



LOD: 0
Tamaño de la pantalla actual: 0,955045
Triángulos: 1 387 109
Vértices: 1 338 250
Canales UV: 2
Tamaño aproximado: 5 773x5 610x5 971
Num Collision Primitives: 1

Figure 3. Unreal Engine Mesh Editor.

3.5 Merge trunk/foliage

In this last phase, the previously digitized trunk and leaves are aligned and merged into a 3D object. The "merge actors" option is used to preserve the textures of both the trunk and the leaves despite converting them into a single mesh.

4. RESULTS

4.1 Case study

The proposed method was tested on four real case studies. The trees located in Palencia (Spain) were acquired with LYNX Mobile Mapper of Optech and the tree located in Monçao (Portugal) was acquired with a Terrestrial Laser Scanner (TLS) Faro Focus X330. Trees were segmented manually form MLS and TLS point clouds.

Common name	Scientific name	No. points	LiDAR
Pine	Pinus Cembra	66k	MLS
Cypress	Cupressus Sempervirens	20k	MLS
Plane tree	Platanus occidentalis	15M	TLS
Poplar	Populus Nigra	305k	MLS

Table 1. Case studies.

4.2 Geometric segmentation

For dense canopy trees, the most relevant geometric features and thresholds identified for trunk segmentation were (Figure 4):

- Verticality: range 0.82 to 1
- Sphericity: range 0 to 0.30
- Planarity: range 0.35 to 0.92
- Eigentropy: range 0.22 to 0.51

For non-dense canopy trees, the most relevant geometric features and thresholds identified for trunk segmentation were (Figure 5):

- Linearity: range 0 to 0.57
- Second principal component: range 0.29 to 0.5
- Verticality: range 0.04 to 1

This trunk/leaf segmentation approach showed a correct behaviour, although due to the limited number of case studies, it is assumed that the performance is not as high as other state-of-the-art methods.

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-555-2023

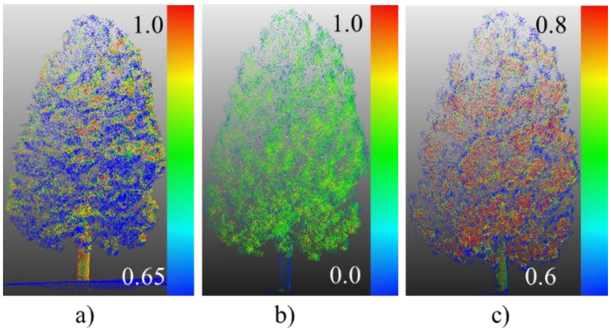


Figure 4. Geometric features for dense tree canopy segmentation: a) verticality, b) scattering, and c) eigentropy.

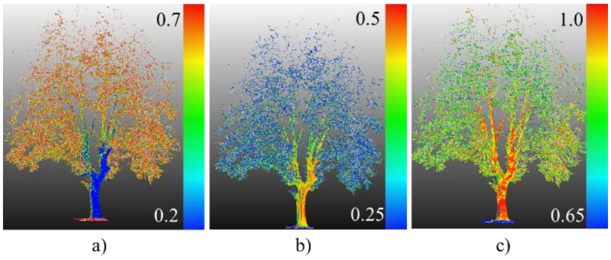


Figure 5. Geometric features for non-dense tree canopy segmentation: a) linearity, b) second principal component, and c) verticality.

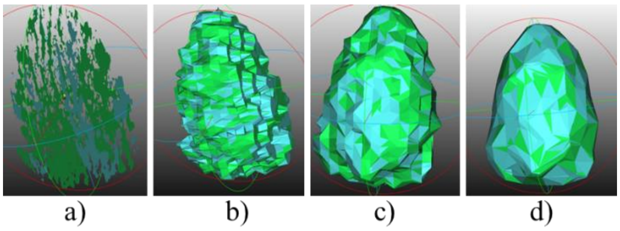


Figure 6. Influences of α on alpha shape generation: a) α = 0.1, b) α = 0.3, c) α = 0.9, d) α = 3.5.

4.3 Alpha shape generation

The selection of α is fundamental to obtain a 3D volume that faithfully fits the dense tree canopy. The parameter α was selected experimentally according to realism of 3D canopies envelopes. Figure 6 shows the envelopes generated according to different α for case study 1. As can be seen, for very low alpha values, a complete envelope was not generated, while higher values fit the shape of the tree. Alpha was set to 3.5 for the two dense canopy trees.

4.4 Tree digitization

The tree digitization results for the four case studies and their comparison with Google Street view images and acquired point clouds are shown below. The pine tree (Figure 7) is the first example of dense canopy tree. The alpha shape allowed the reconstruction with an accurate shape. However, as no digital models of specific pine leaves were available, the filling (geometric terms and colour) of the crown does not match reality.

As the cypress tree (Figure 8) had no visible trunk, the foliage/trunk segmentation was not performed. The digital model responds to the shape of the tree accurately. In this case the alpha shape algorithm was very robust, as the back of the tree was not represented in the input data due to the high density of leaves and MLS trajectory. The colours of the tree did not correspond faithfully to reality, although they could not be obtained from the point cloud either. On the other hand, the model does not contain the seeds of the tree either.

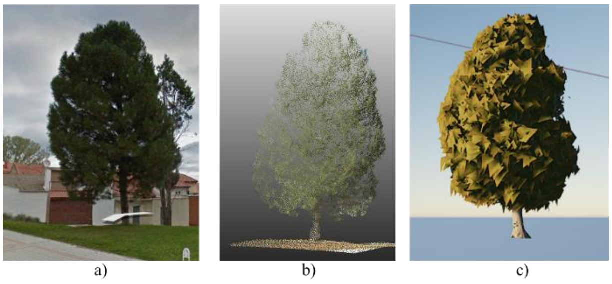


Figure 7. Pine visualized as a Google Street View image (a), point cloud in Cloud Compare (b) and 3D object model in Unreal Engine (c).

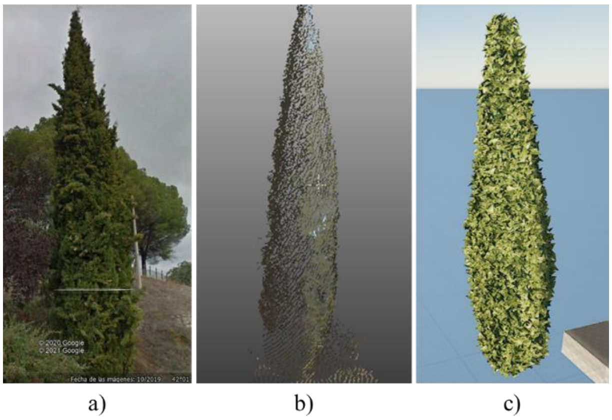


Figure 8. Cypress visualized as a Google Street View image (a), point cloud in Cloud Compare (b) and 3D object model in Unreal Engine (c).

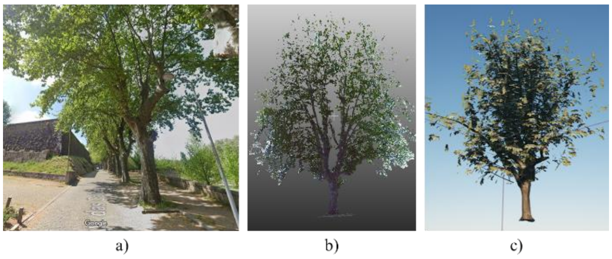


Figure 9. Plane tree visualized as a Google Street View image (a), point cloud in Cloud Compare (b) and 3D object model in Unreal Engine (c).

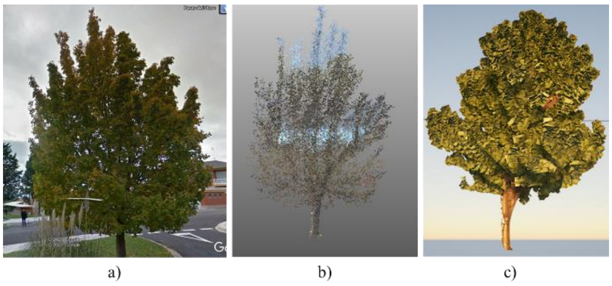


Figure 10. Poplar visualized as a Google Street View image (a), point cloud in Cloud Compare (b) and 3D object model in Unreal Engine (c).

The platanus branches (Figure 9) provided the most complex environment of all case studies. This point cloud acquired with a TLS had a much higher point density than the others. Although the BPA algorithm managed to preserve the geometry of the trunk and branches accurately, the textures were lost and replaced in the process. The result of the digitization was very close to reality. In addition, the leaf pattern used was very similar to the leaves of the platanus.

The poplar tree (Figure 10) shows a very realistic result, better than the point cloud which had colouring errors and few leaves. In addition, the lighting generated by the graphics engine, together with the colour of the leaf model, enhances the realism of the tree.

5. DISCUSSION

The proposed method showed realistic results considering the complexity and variety of trees and LiDAR systems. Trunks and branches were modelled with detailed geometry, although colour and texture were replaced. This means that no colouring errors are transmitted from point clouds to digital models in Unreal Engine, but also a dealignment with reality. The same is true for leaves. Unreal's foliage and mesh editors made it possible to generalize digital leaf models and adjust them faithfully to the tree according to the distribution of the real leaves. Such models are largely dependent on the availability of leaf models according to the tree species, as (Kopf et al., 2008). Although in future work it will be possible to generate a library of specific leaf models.

The proposed method provides a valid alternative to other currently used methods (Runions et al., 2007). The difference and novelty of this procedure in comparison with methods developed in similar research is that, with an optimal LiDAR scan, branches and trunk of the digital tree are identical to the original tree without the application of mathematical algorithms to calculate the probability of growth of the branches. This method obtains the trunk and canopy directly from the point cloud, preserving original shape, size and orientation.

The processing of the trees was almost entirely manual in the proposed method, except for automated algorithms such as BPA, Delaunay triangulation or alpha shape generation. This work presented a conceptualization of the proposed method, and several of the steps will be automated in future work. The identification of the tree species is a state-of-the-art process using Artificial Intelligence (Hartling et al., 2021; Uryasheva et al., 2022; Weiss et al., 2010), as well as trunk/leaf segmentation (Giménez-Gallego et al., 2019; Majeed et al., 2018; Uryasheva et al., 2022). Once the tree specie is known, customized leaf models and segmentation parameters can be automatized and optimized.

The proposed method was independent of whether the input data came from TLS or MLS sensors, as long as the tree point cloud was as complete as possible. The main influence of the input data was on the processing times (Table 2), since the TLS point cloud contained more points, so more triangles were generated in the trunk mesh. The method was implemented in Unreal Engine 427.15, with CPU intel i5-7300HQ 2.5GHz, 8GB RAM, Nvidia GTX 680.

Common name	No. points	LiDAR	Processing time	No. triangles
Pine	66k	MLS	3 s	1386
Cypress	20k	MLS		
Plane tree	15M	TLS	248 s	625168
Poplar	305k	MLS	76 s	38324

Table 2. BPA processing time.

6. CONCLUSSION

In this work, a new method developed for tree digitalisation is presented based on Cloud Compare for the segmentation of the tree parts, Python for the meshing of the surfaces and Unreal Engine to create leaves and texture the tree. The system was tested with four different trees scanned with TLS and MLS systems. Very realistic and detailed results were obtained considering the computational limitations of the equipment, which did not allow obtaining higher quality textures or the implementation of the method in Unreal Engine 5.

This method achieved the successful segmentation of trunk and leaves using the geometric features of the point cloud. This work also proposes for the first time the use of Unreal Engine's "foliage editor" to digitalise trees from point clouds. Starting from a static mesh of a leaf, leaves can be propagated around cylindrical objects, in this case the trunk, being able to control orientation and the range of alignment of these leaves regarding trunk surfaces.

Unreal Engine was a very complex tool for digitising trees from point clouds. Using the mesh and foliage editors it is possible to import and modify pre-generated models of tree parts. Also, illumination and texture options of Unreal Engine give a great realism to the created models.

The results obtained are highly dependent on pre-existing libraries of leaves and canopies. This was visible in the digitisation of dense tree canopies, where the available leaf models did not match the tree species. For non-dense crown trees, the result through the foliage editor was very positive. It can be expected that by generating species-specific leaf and canopy models, results will be significantly improved.

For future work will focus on experiment with more tree samples and species scanned with other LiDAR systems. In addition, full automation of the process will be achieved in the adjustment of leaves to the trunk or the detection of thresholds for trunk/leaf segmentation.

ACKNOWLEDGEMENTS

This research has received funding from Xunta de Galicia through human resources grant (ED481B-2019-061), GAIN (grant number ED431F 2022/08), and Project 4Map4Health, selected in the call ERA-Net CHIST-ERA IV (2019), and founded by the State Research Agency of Spain (reference PCI2020-120705-2/AEI/10.13039/501100011033).

The work presented in this publication is connected to J.B.'s current involvement in the COST Action UB3Guard (CA20132), supported by COST (European Cooperation in Science and Technology)

REFERENCES

Balado, J., Fernández, A., González, E., & Díaz-Vilariño, L. (2023, February). Semantic Point Cloud Segmentation Based on Hexagonal Klemperer Rosette and Machine Learning. In *Advances in Design Engineering III: Proceedings of the XXXI INGEGRAF International Conference* (pp. 617-629). Cham: Springer International Publishing.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. & Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, *5*(4), 349–359.

Cheng, Z.-L., Zhang, X.-P. & Chen, B.-Q. (2007). Simple reconstruction of tree branches from a single range image. *Journal of Computer Science and Technology*, *22*(6), 846–858.

Delaney, B., 2016. Virtual Reality 1.0–The 90's: The Birth of VR in the pages of CyberEdge Journal. CyberEdge Information Services.

Gardner, J. V, Warner, T., Nellis, M. D. & Brandtberg, T. (2003). Virtual reality technology for lidar data analysis. *Geo-Spatial and Temporal Image and Data Exploitation III*, *5097*, 48–57.

Giménez-Gallego, J., González-Teruel, J. D., Jiménez-Buendía, M., Toledo-Moreo, A. B., Soto-Valles, F. & Torres-Sánchez, R. (2019). Segmentation of multiple tree leaves pictures with natural backgrounds using deep learning for image-based agriculture applications. *Applied Sciences*, *10*(1), 202.

Girardeau-Montaut, D., 2016. CloudCompare. France: EDF R&D Telecom ParisTech 11.

Guan, H., Li, J., Cao, S., Yu, Y., 2016. Use of mobile LiDAR in road information inventory: a review. International Journal of Image and Data Fusion 7, 219–242. https://doi.org/10.1080/19479832.2016.1188860

Hartling, S., Sagan, V. & Maimaitijiang, M. (2021). Urban tree species classification using UAV-based multi-sensor data fusion and machine learning. *GIScience & Remote Sensing*, *58*(8), 1250–1275.

Johnson, D.J., Magee, L., Pandit, K., Bourdon, J., Broadbent, E.N., Glenn, K., Kaddoura, Y., Machado, S., Nieves, J., Wilkinson, B.E., Almeyda Zambrano, A.M., Bohlman, S.A., 2021. Canopy tree density and species influence tree regeneration patterns and woody species diversity in a longleaf pine forest. Forest Ecology and Management 490, 119082. https://doi.org/10.1016/j.foreco.2021.119082

Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M. & Lischinski, D. (2008). Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics (TOG)*, *27*(5), 1–10.

Livny, Y., Pirk, S., Cheng, Z., Yan, F., Deussen, O., Cohen-Or, D. & Chen, B. (2011). Texture-lobes for tree modelling. *ACM Transactions on Graphics (TOG)*, *30*(4), 1–10.

Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H. & El-Sana, J. (2010). Automatic reconstruction of tree skeletal structures from point clouds. In *ACM SIGGRAPH Asia 2010 papers* (pp. 1–8).

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q. & Whiting, M. D. (2018). Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. *IFAC-PapersOnLine*, *51*(17), 75–80.

Pielou, E. C. (1966). The measurement of diversity in different types of biological collections. *Journal of Theoretical Biology*, *13*, 131–144.

Quixel Bridge. (n.d.). Retrieved March 16, 2023, from https://quixel.com/bridge

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. Remote Sensing 5, 491–520. https://doi.org/10.3390/rs5020491

Reche-Martinez, A., Martin, I. & Drettakis, G. (2004). Volumetric reconstruction and interactive rendering of trees from photographs. In *ACM SIGGRAPH 2004 Papers* (pp. 720–727).

Runions, A., Lane, B. & Prusinkiewicz, P. (2007). Modeling trees with a space colonization algorithm. *Nph*, *7*(63–70), 6.

Sanders, A. (2016). *An introduction to Unreal engine 4*. AK Peters/CRC Press.

Shutterstock. (n.d.). *Turbosquid*. https://www.turbosquid.com/es/?&utm_source=google&utm_medium=cpc&utm_campaign=ES-es-TS-Brand&utm_content=Brand-TurboSquid&utm_term=turbosquid&mt=e&dev=c&itemid=&targid=kwd-297496938642&loc=1005518&ntwk=g&dmod=&adp=&gclid=CjwKCAiA3pugBhAwEiwAWFzwddBTxtvvZsr66mZfAWdgOqWp86tQCui67We__bFuntdiXRou_chmzRoC8hcQAvD_BwE&gclsrc=aw.ds

Simpson, J. E., Smith, T. E. L. & Wooster, M. J. (2017). Assessment of Errors Caused by Forest Vegetation Structure in Airborne LiDAR-Derived DTMs. In *Remote Sensing* (Vol. 9, Issue 11). https://doi.org/10.3390/rs9111101

Št'ava, O., Beneš, B., Měch, R., Aliaga, D. G. & Krištof, P. (2010). Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum*, *29*(2), 665–674.

Tan, P., Zeng, G., Wang, J., Kang, S. B. & Quan, L. (2007). Image-based tree modeling. In *ACM SIGGRAPH 2007 papers* (pp. 87-es).

Uryasheva, A., Kalashnikova, A., Shadrin, D., Evteeva, K., Moskovtsev, E. & Rodichenko, N. (2022). Computer vision-based platform for apple leaves segmentation in field conditions to support digital phenotyping. *Computers and Electronics in Agriculture*, *201*, 107269.

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

Weiss, U., Biber, P., Laible, S., Bohlmann, K. & Zell, A. (2010). Plant species classification using a 3D LIDAR sensor and machine learning. *2010 Ninth International Conference on Machine Learning and Applications*, 339–345.

White, J.C., Coops, N.C., Wulder, M.A., Vastaranta, M., Hilker, T., Tompalski, P., 2016. Remote Sensing Technologies for Enhancing Forest Inventories: A Review. Canadian Journal of Remote Sensing 42, 619–641. https://doi.org/10.1080/07038992.2016.1207484

Xu, H., Gossett, N. & Chen, B. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics (TOG)*, *26*(4), 19-es.