# Three-Dimensional Modeling and Visualization of Single Tree LiDAR Point Cloud Using Matrixial Form

Fayez Tarsha Kurdi, Elżbieta Lewandowicz, Jie Shan, and Zahra Gharineiat

***Abstract*—Tree modeling and visualization still represent a challenge in the light detecting and ranging area. Starting from the segmented tree point clouds, this article presents an innovative tree modeling and visualization approach. The algorithm simulates the tree point cloud by a rotating surface. Three matrices, *X, Y*, and *Z*, are calculated by considering the middle of the projected tree point cloud on the horizontal plane. This mathematical form not only allows tree modeling and visualization but also permits the calculation of geometric characteristics and parameters of the tree. The superimposition of the tree point cloud over the constructed model confirms its high accuracy where all the points of the tree cloud are within the constructed model. The tests with multiple single trees demonstrate an overall average fit between 0.3 and 0.89 m. The built tree models are also compliant with the Open Geospatial Consortium CityGML standards at the level of a physical model. This approach opens a door to numerous applications for visualization, computation, and study of forestry and vegetation in urban as well as rural areas.**

***Index Terms*—Light detection and ranging (LiDAR), Open Geospatial Consortium (OGC) CityGML physical model, tree model, vegetation, visualization.**

## I. Introduction

LIGHT detection and ranging (LiDAR) technology can be used in both urban and rural forests [1], [2]. In this context, static terrestrial scanning can play a major role in mapping and modeling for small-scale projects [3], [4]. Some mapping and modeling tasks require airborne scanning where the unmanned aerial vehicle (UAV) is widely employed regarding easy operation and cheap cost compared with the conventional airplanes, although the regulations of UAV flight and mapping specifications are still a research topic [5], [6] and may vary among different applications. Nevertheless, whichever LiDAR data measurement tool is employed, one of the first steps of data processing is data classification [7], [8], [9], i.e., to obtain mostly terrain, vegetation, and buildings classes from the source point clouds. In the recent years, machine learning techniques have been widely used to automatically classify [10], [11] and model [12] the LiDAR data.

Concerning the vegetation class, trees are often the main component of this class in urban and suburban areas. Regardless of the tree species, a tree consists of roots, a trunk, branches, leaves, and maybe fruits. According to the season of data acquisition and the tree species, the leaf mass can be present or not. However, the tree shape can be depicted according to their generic forms, such as spreading, columnar, globe, fan, broad oval, and narrow conical [13]. From another viewpoint, the geometric form can be semiregular, such as pine trees, or irregular acacia trees. The problem with tree modeling is that trees do not have a regular geometric form, such as manmade objects, e.g., buildings and bridges. Moreover, most trees grow and change their shapes during the time and seasons, which is why most tree modeling approaches in the literature focus on extracting the tree parameters, such as trunk diameters, height, and biomass volume. To best represent and model trees, the Open Geospatial Consortium (OGC), an international organization developing and promoting open standards for geospatial data and services, facilitating interoperability in the geospatial domain, has endorsed CityGML as an adopted standard for representing three-dimensional (3-D) urban models [14]. Although CityGML has simplified 3-D tree symbols, adapted to tree parameters, it does not have tools for the automatic creation of single 3-D tree models that can precisely present the actual shape of real trees. Such models would correspond to physical models. As such, tree model reconstruction and visualization from the LiDAR point cloud still pose a challenge.

The main contribution of this article is the introduction of a 3-D parameterizable and visualizable mathematical model for single tree point clouds. To construct this model, a rotating surface is adapted to simulate the tree geometry. To be specific, three matrices *X, Y*, and *Z* are formed from a tree point cloud to represent the tree. This model uses a vertical cylinder of zero radius passing through the middle of the bounding box

Manuscript received 4 September 2023; revised 22 October 2023 and 19 December 2023; accepted 22 December 2023. Date of publication 4 January 2024; date of current version 16 January 2024. This work was supported in part by the Statutory Research Project of the Faculty of Geoengineering of the University of Warmia and Mazury in Olsztyn, Poland, entitled "Geoinformation from the Theoretical, Analytical and Practical Perspective" under Grant 29.610.008-110. *(Corresponding author: Fayez Tarsha Kurdi.)*

Fayez Tarsha Kurdi and Zahra Gharineiat are with the School of Surveying and Built Environment, University of Southern Queensland, Springfield Campus, Springfield, QLD 4300, Australia (e-mail: fayez.tarshakurdi@usq.edu.au; zahra.gharineiat@usq.edu.au).

Elżbieta Lewandowicz is with the Department of Geoinformation and Cartography, Institute of Geodesy and Civil Engineering, Faculty of Geongineering of the University of Warmia and Mazury in Olsztyn, 10-719 Olsztyn, Poland (e-mail: leela@uwm.edu.pl).

Jie Shan is with the School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: jshan@purdue.edu).

Digital Object Identifier 10.1109/JSTARS.2024.3349549

defined by the tree point cloud. In this way, every LiDAR point corresponds to one cell in the model matrices. This solution strategy enables us to build an innovative algorithm that can automatically construct a tree model from the LiDAR point cloud. Our development has the following advantages.

1) The tree model is saved in a mathematical form.
2) The suggested algorithm allows for visualizing the tree model.
3) The tree model allows calculating the tree parameters.
4) The tree model can consider the trunk and the crown according to their geometric forms.
5) The tree model corresponds to the physical model according to the OGC CityGML standard.

To summarize, the motivation for developing the presented approach in this article is to construct a tree modeling algorithm that is able to consider the general case of a tree point cloud regardless of the complexity level of its geometric form. The envisaged tree model should be simple, visualizable, and allow calculating the tree parameters. Hence, the novelty of this article is given as follows.

1) Introduction of an innovative modeling algorithm that is adaptive to the geometric complexity of the tree.
2) The developed algorithm is based on a matrixial presentation and, thus, can accommodate different point densities.
3) The proposed algorithm is easy to implement and only needs minimal memory and computation cost.

In this study, the authors do not intend to obtain a tree skeleton or a real model of a tree consisting of a trunk, branches, leaves, and crown, which is the closest to reality. Such solutions have already been proposed. Instead, the intention is to have a simple model that can be rapidly created from LiDAR measurement and is convenient for visualization and computation. It is supposed to present a physical object model [14] of a tree compliant with the OGC CityGML3 standard.

The rest of this article is organized as follows. Section II makes a literature review of similar studies carried out before. Section III presents the employed LiDAR dataset in this study, followed by Section IV elaborating the suggested approach and illustrating the obtained results in Section V. Finally, Section VI concludes this article.

## II. Literature Review

The data collected with LiDAR aerial mapping technology, terrestrial mapping tools, backpack mobile mapping systems, and smartphones are used to model 3-D geographic space and capture changes in terrain features. Data points can be classified into thematic subsets relating to landform, land development [15], and tall and short vegetation [16]. These tools create new research opportunities by converting point clouds to subsets representing individual topographic objects and transforming them into vector objects [16], [17]. The international standards for 3-D modeling are set by the OGC CityGML model [14], which defines the principles for mapping physical topographic objects and physical volumetric objects [14], [18]. Physical volumetric models are combined with semantic and functional data to develop the detailed models of the urban systems [18], [22], road networks, and other systems, including in real time [23], [24].

Most 3-D models are developed to represent cities and urban settings [19], [25]. Such modeling has been automated at the Level of Details 2 [26], [29]. Forests are also mapped with the use of 3-D models [30], [32]. Various methods for extracting individual trees have been proposed in the pieces of literature [33], [35]. These techniques are used to model urban green areas, roadside trees [36], and individual dendrological objects. LiDAR input datasets have been expanded to include topographic data collected with the use of various platforms, including vehicles, backpack mobile mapping systems, and stationary devices [37], [39]. The generated tree models, in particular tree canopies, have been used to determine shaded areas in street infrastructure [39] and the extent to which trees obstruct light access in agricultural crop fields [40].

According to the European Union (EU) regulations [41], all EU countries need to report on greenhouse gas emissions in forests. The models of above-ground biomass (AGB) are generated for that purpose [42], [44]. Various solutions for modeling forests with the use of machine learning methods and hyperspectral and LiDAR data have been described in the literature [43], [45]. The AGB model can be used to calculate the carbon footprint and the carbon balance [42], [46]. The carbon footprint model is more reliable when it is developed based on the biomass of individual trees [47], [48]. Models of individual forest trees are also generated for research needs on forest management and environmental assessments of habitats of various animal species, including birds [49].

Data subsets describing individual trees can be extracted from LiDAR data [38], [50], [52] for modeling purposes. Tree models are generated from LiDAR point clouds with the use of clustering algorithms in machine learning that extract the geometric properties of tree trunks, branches, and canopies [53], [60]. Solutions for modeling individual leaves have also been proposed [54], [57]. Voronoi diagrams have been used to extract individual trees from point clouds, and tree canopies have been extracted with region growing algorithms and grouping algorithms [38]. Zhu et al. [55] concluded that the models of the structure and shape of trees, branches, and canopies are not always satisfactory in terms of silhouette and detail. The cited authors proposed an approach to 3-D tree canopy reconstruction by developing a concave mesh from all scanned points.

Individual trees can also be modeled with the use of selected point-cloud layers (cross sections) [56]. The model is generated based on the distribution of leaf area across layers. In a study by Reckziegel et al. [40], horizontal cross sections at different heights were applied to develop a canopy height model in predefined steps. Local maxima from the subsets of LiDAR points were generated in successive layers to develop a 3-D model of the tree trunk. Trees are increasingly modeled based on the topological neighborhoods between subsets of LiDAR points that present trunk and branches. Graphs and optimization algorithms are applied to model tree trunks and branches [61], [67]. These tools and methods are also used to develop hierarchical models of trees [58] and to transform skeleton models at various levels of detail [59].

Similar to building information modeling, tree information modeling is a digital representation of the physical and functional characteristics of a tree at different levels of detail based on the CityGML 3.0 standard [20]. The models of individual trees are generated at different levels of detail by researchers [68], [69]. A review of the literature indicates that methods for automated modeling of individual trees based on laser scanning data should be developed for various applications. Two main approaches to 3-D trees modeling dominate. The first approach is based on the generated skeletons [56], [61], [65]. It is proposed to supplement it with leaves using an algorithm based on the clump of leaves concepts [53]. This solution requires the recording of geometry and semantic data of the skeleton in graph structures [56], [66]. Leaves should be modeled only on growth twigs. The second approach involves dividing the tree point cloud into horizontal slices, then extracts the circumference of each slice. From all extracted circumferences, it defines the tree crown and trunk [40], [59], [60], [70].

The aim of this study is to develop an algorithm for automatic tree model generation from LiDAR points representing individual trees. The generated model will be a physical volumetric model of the space occupied by the tree. Unlike the conventional raster or voxel data models, our development is based on both metric and angular divisions. The cells in such a division are stored in a regular matrix format. As a result of this model, the original irregular LiDAR points of a tree are regularized or structuralized. This simplification facilitates not only the visualization and presentation of trees but also eases the derivation of tree properties for canopy change monitoring during growth and degradation.

## III. Datasets

Subsets of airborne laser scanning data were obtained for the study from the polish spatial data infrastructure. LiDAR measurements were conducted in March 2018 with a density of 12 points/m$^2$. The acquired data represent individual trees on the campus of the University of Warmia and Mazury in Olsztyn, Poland. The subsets of LiDAR points representing various species and genera of deciduous and coniferous trees were selected (see Fig. 1) for the study based on a list of dendrological objects in the geographic information system database created in 2012 based on classic field surveying data [60]. It is desirable to update the dendrological database from 2012 and supplement it with physical 3-D tree models, specified on the LiDAR data. In the following sections, such a solution is presented based on the selected subsets of the LiDAR point cloud of single trees.

## IV. Tree Matrices

In the context of 3-D model construction of the scanned scene and regarding the high diversity and complexity of tree geometry, as well as the nonstopped demand for an algorithm that can model and visualize the vegetation in forest and urban areas, this article proposes an innovative tree modeling approach. It simulates the rotating surface suggested by Lewandowicz et al. [29], [72], which was originally suggested for building modeling. This strategy uses the matrixial form to construct the tree model that conserves the tree geometry depicted through the tree LiDAR point cloud. This section will detail the suggested method.

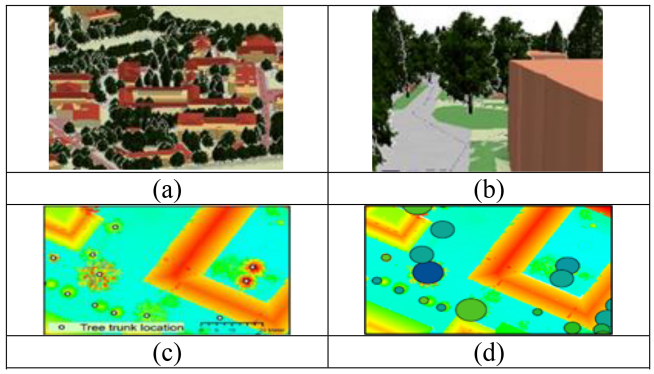


Fig. 1. 3-D visualization of trees on the campus of the University of Warmia and Mazury based on the dendrological database by using (a) Esri symbolization and (b) Own tree symbolization. (c) Location of tree trunks and (d) diameters of tree canopies in the dendrological database based on the field measurements, with LiDAR data classified based on height [71].

First, the middle of the bounding box of the projected point cloud on the horizontal plane (*OXY*) is calculated. The underlying assumption is that the *Z*-axis of the coordinate system is vertical and the *XY*-plane is horizontal. As such, this point can represent the point-cloud gravity center where the trees geometric form is symmetrical, and the LiDAR point distribution is homogeneous. The equations suggested by Lewandowicz et al. [29] are employed as follows:

$$X_g = X + \frac{X_{\max} - X_{\min}}{2}, Y_g = Y + \frac{Y_{\max} - Y_{\min}}{2} \quad (1)$$

where $X_g$ and $Y_g$ are the coordinates of the middle of the bounding box; and $X_{\min}$, $Y_{\min}$, $X_{\max}$, and $Y_{\max}$ are the extreme values of $X$ and $Y$ coordinates (minimum and maximum).

Second, a vertical line is introduced that passes through the calculated center of the bounding box to represent a cylinder having a zero radius. This cylinder represents a rotating surface that can be expressed by three matrices (2)–(6) shown at the bottom of the next page. Indeed, this 3-D surface can be presented through three matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$. The first two matrices $\mathbf{X}$ and $\mathbf{Y}$ define a grid in the $OXY$-plane, whereas the third matrix $\mathbf{Z}$ represents the heights of this grid. Suppose a 3-D rotating surface around the $Z$-axis, this surface is sliced by horizontal planes into a series of layers. Every slice represents a row in the three matrices. Since all points of one slice have the same $Z$ value, one-row cells in the $\mathbf{Z}$ matrix have the same value $Z_i$. Furthermore, in the $\mathbf{X}$ and $\mathbf{Y}$ matrices, one slice represents a circle included or parallel to the plane $OXY$ because the supposed surface is the rotating surface around the $Z$-axis. If one circle is divided radially into *t* circular sectors, each sector will present one cell among that row. Hence, the coordinates of this cell can be calculated regarding the farthest west point, the circle radius, and the angle between the $X$-axis and the line connecting the concerning point and the gravity center (see Fig. 2). In this context, matrices 2–4 can be written using the generic form (5).

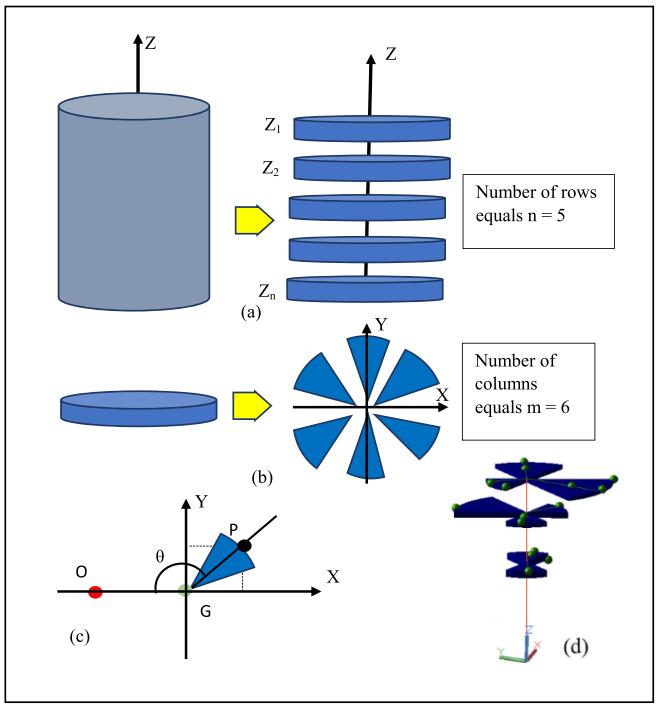


Fig. 2. Formation of the rotating-surface matrices. (a) Division of the surface by horizontal planes into "$n = 5$" slices. (b) Radialy division of one slice into "$t = 6$" circular sectors. (c) Calculation of the $\mathbf{X}$ and $\mathbf{Y}$ matrix cell values, $G$ is the gravity center, $P$ is a LiDAR point, $O$ is the calculation origin, and (d) LiDAR points lie on the 3-D tree model.

Fig. 2 illustrates how the rotating surface, which is taken as a cylinder, is divided into horizontal slices, where each slice has a constant $Z$ coordinate value. Thereafter, each slice is divided into a given angular sector. Finally, each angular sector allows the calculation of the $X$ and $Y$ matrix cell values depending on the rotation origin ($O$), the point-cloud gravity center ($G$), and the circle radius.

The dimension of the $\mathbf{Z}$ matrix is $n \times m$, and the values of each row are identical ($n$ and $m$ are explained in the following text). $X_g$ and $Y_g$ are the coordinates of the gravity center (1); $X_i$, $Y_i$, and $Z_i$, $(i = 1, \ldots, n)$ are the point coordinates of the sectors; $j = 1, \ldots, m$; $n$ is the number of points in the semicross section. The three matrices $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$ have the same dimensions. $\alpha_i$ and $\beta_i$ are the step values of $X$ and $Y$, respectively; and $m$ is the number of columns in matrix $X$.

To create these matrices, the LiDAR tree point cloud must first be sorted descending according to the $Z$ coordinate values. The repeated $Z$ coordinate values shall be eliminated. The number of columns can be considered according to the point density and must meet $m = 4k + 1 > 7$, where $k$ is a given value representing the number of considered sectors by slice quarter.

In this article, considering all tree point clouds having approximately the same point density lower than 10 points/m$^2$, then the number of columns is taken equal to ($m = 25$ for $k = 6$). As an example, to understand how the number of columns ($m$) is calculated, when $k = 1$, it is noted that one circular slice consists of four quarters, which can represent four columns [$C1$ $C2$ $C3$ $C4$] of the model matrices. To make the rotating surface closed, the first angular sector must be added again after the last

$$\boldsymbol{X} = \begin{bmatrix} X_g X_g + \beta_{1,1} & X_g + 2\beta_{1,2} & \ldots\ldots & X_g + (m-1) \times \beta_{1,m-1} \\ X_g X_g + \beta_{2,1} & X_g + 2\beta_{2,2} & \ldots\ldots & X_g + (m-1) \times \beta_{2,m-1} \\ X_g X_g + \beta_{3,1} & X_g + 2\beta_{3,2} & \ldots\ldots & X_g + (m-1) \times \beta_{3,m-1} \\ \cdot & \cdot & \cdot & \cdot \\ X_g\ X_g + \beta_{n,1} & X_g + 2\beta_{n,2} & \ldots\ldots & X_g + (m-1) \times \beta_{n,m-1} \end{bmatrix} \tag{2}$$

$$\boldsymbol{Y} = \begin{bmatrix} Y_1 & Y_1 + \alpha_{1,1} & Y_1 + 2\alpha_{1,2} & \ldots. & Y_1 + (m-1) \times \alpha_{1,m-1} \\ Y_2 & Y_2 + \alpha_{2,1} & Y_2 + 2\alpha_{2,2} & \ldots. & Y_2 + (m-1) \times \alpha_{2,m-1} \\ Y_3 & Y_3 + \alpha_{3,1} & Y_3 + 2\alpha_{3,2} & \ldots. & Y_3 + (m-1) \times \alpha_{3,m-1} \\ \cdot & \cdot & \cdot & & \cdot \\ Y_n & Y_n + \alpha_{n,1} & Y_n + 2\alpha_{n,2} & \ldots. & Y_n + (m-1) \times \alpha_{n,m-1} \end{bmatrix} \tag{3}$$

$$\boldsymbol{Z} = \begin{bmatrix} Z_1 & Z_1 & \cdots & Z_1 \\ Z_2 & Z_2 & \cdots & Z_2 \\ \vdots & \vdots & \vdots & \vdots \\ Z_n & Z_n & Z_n & Z_n \end{bmatrix} \tag{4}$$

$$\left.\begin{array}{c} X_{i,j} = X_g + (j-1) \times \beta_{i,j-1} \\ Y_{i,j} = Y_i + (j-1) \times \alpha_{i,j-1} \\ Z_{i,j} = Z_i \\ \text{where } i: 1 \text{ to } n; j: 1 \text{ to m} \end{array}\right\} \tag{5}$$

$$\alpha_{i,j} = (Y_g - Y_i) \sin\left(\frac{2j\pi}{m} + \frac{3\pi}{2}\right), \beta_{i,j} = (Y_g - Y_i) \cos\left(\frac{2j\pi}{m} + \frac{3\pi}{2}\right). \tag{6}$$

column, and the number of columns becomes 5 [$C1$ $C2$ $C3$ $C4$ $C1$]. Furthermore, the representation of one slice quarter by only one point ($K = 1$) will simplify the tree model and produces an undesirable deformed model. That is why, a decision is taken to consider the minimum number of angular sectors of each slice quarter equals $k = 2$ instead of 1. Hence, for two (2) sectors by quarter ($k = 2$), there will be $m = 4 \times 2 + 1 = 9$ columns; for three sectors by quarter ($k = 3$), there will be 13 columns; and for six sectors by quarter ($k = 4$), there will be 25 columns. At this stage, an important question arises: how many angular sections must be considered by slice quarter? (That means, what is the considered value of the parameter $k$?). The answer to this question is obviously related to the LiDAR point density. Indeed, when the point density is high, extra slice divisions are requested to minimize the number of points represented by each sector. At this stage, more investigation is requested in future work to determine the exact equation that calculates the number of columns as a function of point density.

The following equation is used to form the three matrices:

$$X_{i,j} = X_g \;\; Y_{i,j} = Y_g \;\; Z_{i,j} = \text{cloud}_{i,3} \tag{7}$$

where $\text{cloud}_{i,3}$ is the corresponding $Z$ coordinate in the cloud point of row number $i$ (where the point cloud was previously sorted descending according to the $Z$ coordinate values).

Third, considering the point cloud before removing the redundant $Z$ coordinate value points, each point will lead to modifying one cell in the $X$ matrix as well as the same corresponding cell in the $Y$ matrix. The row of the focus cell is determined from the $Z$ coordinate value of the point. Hence, to determine the column of the concerning cell, it is necessary to calculate the $\theta$ angle with (8), which is formed between the $X$-axis and the line connecting the concerning point and the gravity center. In fact, the angle $\theta$ depicts the location of the concerning point regarding the gravity center

$$\theta = \arctan \frac{\text{abs}\,(\Delta \text{Y})}{\text{abs}\,(\Delta X)} = \arctan \frac{\text{abs}\,(Y - Y_g)}{\text{abs}\,(X - X_g)}. \tag{8}$$

Once $\theta_o$ is calculated, the corresponding cell that will be modified can be calculated according to the following equation:

$$\text{CN} = m \times \text{round}\left(\frac{\theta_o}{2\pi}\right) \tag{9}$$

where CN is the column number in matrices **X** and **Y** (2) and (3), $m$ is the number of columns in matrix **X**, and "round" is a function that provides the round value of a given real number. The new value of the corresponding cells in **X** and **Y** matrices is calculated using the following equation:

$$\left.\begin{aligned} X_{\text{CN}} &= X_g + \text{Dis}_{pg} \times \cos\theta_o \\ Y_{\text{CN}} &= Y_g + \text{Dis}_{pg} \times \sin\theta_o \end{aligned}\right\} \tag{10}$$

where $X_{\text{CN}}$ and $Y_{\text{CN}}$ are the corresponding cell values in the **X** and **Y** matrices, $\text{Dis}_{pg}$ is the distance between the gravity center $g$ and the given LiDAR point $p$, which can be calculated according to the following equation:

$$\text{Dis}_{pg} = \sqrt{(X_p - X_g)^2 + (Y_p - Y_g)^2}. \tag{11}$$

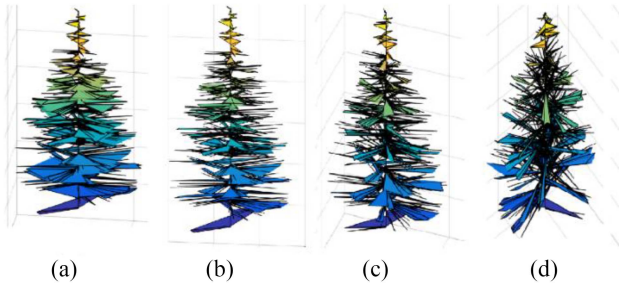


Fig. 3. Visualization of a tree modeled by the model matrices with, respectively, (a) 9, (b) 17, (c) 25, and (d) 33 columns, where the number of rows is 369.

To conclude, the application of (9) to each tree LiDAR point allows filling the corresponding cell in the $X$ and $Y$ matrices. Once all tree points are employed, the model matrices will contain two types of cells: filled cells where their values are calculated from the LiDAR points, and empty cells where there are no available LiDAR points to calculate their values. Hence, the empty cells remaining in the $X$ and $Y$ matrices are filled by $X_g$ and $Y_g$ values, consecutively. In essence, the suggested modeling approach transforms the list of trees LiDAR points defined by their cartesian coordinates $X_i$, $Y_i$, and $Z_i$ into three matrices $X$, $Y$, and $Z$. The new matrixial form of the tree point cloud not only allows visualization of the tree within a 3-D scene but also allows fitting the tree model to improve it and realize the relevant calculation, such as tree height, footprint diameter, and upper biomass volume on the tree cloud (see Section V-C).

## V. Results and Discussion

Once the suggested tree modeling approach has been detailed, we will analyze the obtained tree models and discuss their accuracy.

### A. Analysis of Tree Model Structure

As shown above, the tree point-cloud model consists of three matrices $X$, $Y$, and $Z$. On the one hand, one row of these matrices represents one level of slice. On the other hand, the number of columns of these matrices can be selected arbitrarily, as long as it satisfies $4k + 1 > 7$ or $2k > 3$. This section focuses on the analysis of the model cell values.

If a point cloud is laid on the constructed model, the number of points inside a model cell will be variable according to the cell dimension. Table I compares the number of LiDAR points located inside model cells considering four cases where the number of columns of the model matrices is, respectively, 9, 17, 25, and 33 (see Fig. 3 and Table I). In this context, the model cells can be classified into four categories according to the number of LiDAR points inside the cell. First, it is the case when a model cell contains no LiDAR points, i.e., the cell is empty. It can be noted from Table I that more than 75% of the model cells belong to this category. Moreover, when the number of columns of the model matrices increases, the number of cells in this family doubled. It can be noted from Table I for Tree 1 that when the number of columns of the model matrices equals 9, the number of model cell contains 9353 LiDAR points. However, it

TABLE I
NUMBER OF LiDAR POINTS LOCATED INSIDE MODEL CELLS FOR THREE EXAMPLE TREES

| | Number of matrix columns | Number of model cells with 0,1,2 and >2 lidar points | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | >2 |
| **Tree1** | 9 | 9353 | 2048 | 414 | 108 |
| | 17 | 19 745 | 2398 | 330 | 52 |
| | 25 | 30 229 | 2602 | 262 | 32 |
| | 33 | 40 789 | 2674 | 233 | 29 |
| **Tree2** | 9 | 9317 | 1990 | 357 | 81 |
| | 17 | 19 588 | 2292 | 255 | 50 |
| | 25 | 29 958 | 2409 | 228 | 30 |
| | 33 | 40 365 | 2468 | 210 | 22 |
| **Tree7** | 9 | 2723 | 442 | 27 | 3 |
| | 17 | 5558 | 450 | 26 | 1 |
| | 25 | 8396 | 466 | 18 | 1 |
| | 33 | 11 232 | 466 | 20 | 1 |

becomes 19 745 when the number of columns equals 17. The same notice can be observed when the number of model matrix columns increases to 25 and 33. Second, a model cell contains only one LiDAR point. It can be noted from Table I that the number of cells in this category increases slightly when the number of columns increases. Indeed, increasing the number of columns leads to a reduction on the cell's dimension. Finally, in the third and fourth cases, a cell may have two or more points. In fact, the number of cells in these two categories is negligible due to the popularity of the first two cases.

To conclude, this analysis allowed understand the relationship between the point density and the number of columns in the tree model matrices. Moreover, it also permitted the reader to deeply perceive how the tree point cloud could be transformed into a 3-D model.

Finally, in the automatic modeling of LiDAR data, there are two main approach families (model-driven and data-driven). At this stage, it is important to discuss the location of the suggested approach within these two families. According to Tarsha Kurdi et al. [73], the model-driven approach consists of searching for the most appropriate model among basic models contained in a model library, whereas the data-driven approach attempts to construct a model by using a series of more or less complex operations, allowing the generation of a model without employing a specific library. As the proposed approach does not use a model library, it seems to be nearer to a data-driven than a model-driven approach.

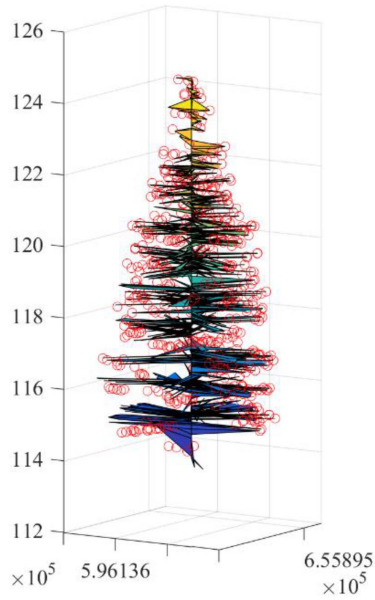


Fig. 4. Superimposition of a tree point cloud (red circles) over its tree model.

From another viewpoint, in both modeling families, a model, such as a 3-D building model, is composed of planes and edges extracted from the LiDAR point cloud. That means the model consists of a group of basic vector geometric elements connected according to predefined topological relationships. The suggested approach uses different modeling concepts based on the mathematical description of the focus object through three matrices, which can be visualized in 3-D space. That is why, the suggested approach may present a new modeling family that will be added to the two main modeling approach families.

### B. Accuracy

The modeling accuracy will be estimated by comparing the constructed tree model with its point cloud [11]. Fig. 4 shows the superimposition of the point cloud over the tree model. It is noticed that the constructed model fits accurately the tree point cloud. At this stage, it is necessary to note that if two or more LiDAR points have the same $Z$ coordinate value and the same angle $\theta$ value, only one of these points will be considered and the other points will be presented by the considered point. To avoid neglecting any points, it is sufficient to increase or decrease their $Z$ coordinate value by only a small number, such as 1 mm. According to Table I, the percentage of neglected points does not surpass 5% of the point cloud regarding the point irregular distribution.

From (2) and (3) as well as Fig. 5, it can be noted that one tree model consists of a matrix of cells connected through robust neighbor relationships. To analyze a tree model's accuracy regarding the tree point cloud, the mean cell dimensions are used as an evaluation metric. The width ($C_W$) and height ($C_H$) of a cell can be calculated using the following equation:

$$C_W = \frac{2\pi \mathrm{Dis}_{pg}}{m}; C_H = Z_i - Z_{i-1} \neq 0 \tag{12}$$

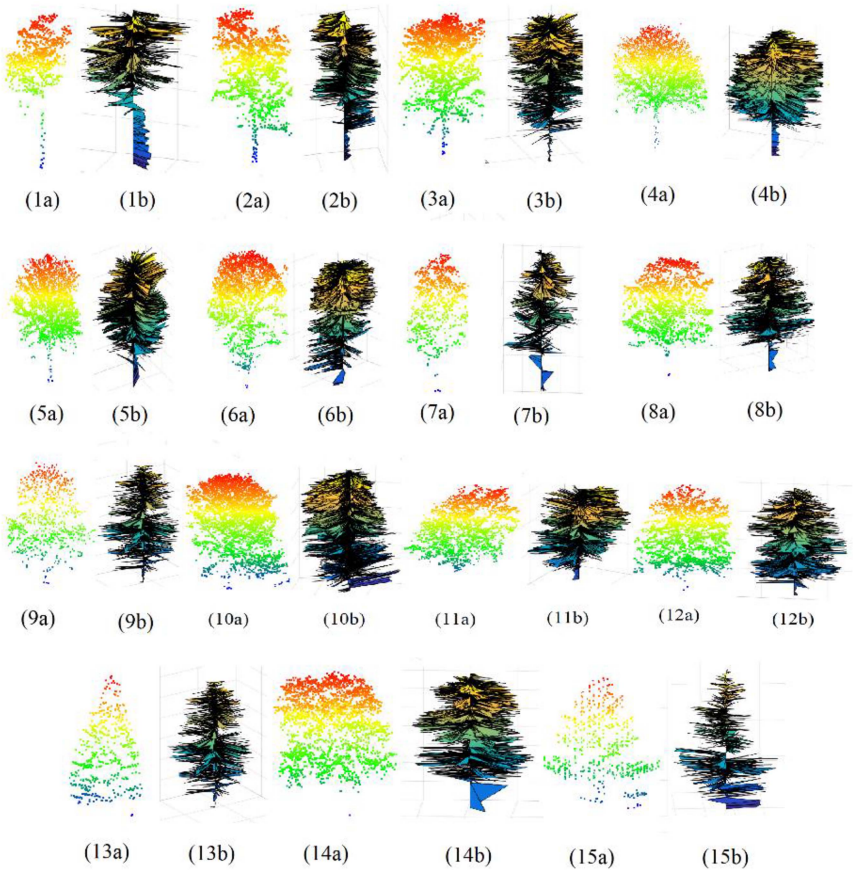


Fig. 5. Visualization of tree point clouds and tree models; (1a) to (15a) are the tree point clouds; and (1b) to (15b) are the derived tree models.

TABLE II
ACCURACY OF THE TREE MODELS FOR $m = 33$ (THE NUMBER OF COLUMNS IN THE MATRICES)

| | Min $C_W$ (m) | Max $C_W$ (m) | Mean $C_W$ (m) | Min $C_H$ (m) | Max $C_H$ (m) | Mean $C_H$ (m) |
|---|---|---|---|---|---|---|
| **Tree 1** | 0.01 | 1.74 | 0.85 | 0.01 | 0.58 | 0.2 |
| **Tree 2** | 0.01 | 1.86 | 0.89 | 0.01 | 0.39 | 0.2 |
| **Tree 7** | 0.01 | 0.63 | 0.36 | 0.01 | 0.51 | 0.3 |

where $\text{Dis}_{pg}$ is the distance between the gravity center $g$ and the given point (11), and $m$ is the number of columns in matrix **X**.

From (12), it can be noted that the cell dimensions are related to the number of columns in the tree model matrices, the distance from the rotating axis, and the point density. Hence, when the number of columns of the model matrices increases, the cell width will decrease. Moreover, $C_W$ and $C_H$ values are variable from one point to another, which is why, for each tree model, the minimum, maximum, and mean values of these parameters can be calculated (see Table II). In fact, the two parameters $C_W$ and $C_H$ also express the accuracy of the cell location within the constructed tree model. Indeed, one LiDAR point is supposed to be located in the cell center, but in practice, it can be located anywhere inside the cell space. To conclude, the accuracy of the tree model can be improved by increasing the number of columns of the model matrices (2)–(4).

### C. Tree Parameter Calculation

In the context of tree modeling starting from LiDAR data, it is necessary to estimate the tree parameters, such as the height, crown diameter, trunk diameter, and the upper biomass volume [44], [54], [66], [67]. Although Herrero-Huerta et al. [67] define seven parameters as a full description of a tree, only three are considered in this article: the tree height, the crown diameter, and the trunk diameter. In fact, this choice has been adopted because the other parameters can be deduced from these three. The tree height (*Ht*) can be calculated directly from (13) using the matrix $Z$ (4), where $Z_1$ and $Z_n$ are the $Z$ coordinates of the highest and lowest points of the tree point cloud. The crown diameter equals the tree footprint diameter (*Fd*), which can be calculated using (14), where the tree footprint is supposed to be circular. Concerning the trunk diameter, it can be noted from Fig. 5 that the trunk LiDAR points may be missed or indistinguishable in the tree point cloud (Trees 6–15) because the laser pulses did not

TABLE III
CALCULATION PERCENTAGE OF TRUNK RADIUS TO TREE FOOTPRINT RADIUS

| Tree number | Fr: Footprint radius (m) | Tr: Trunk radius (m) | Fr/Tr ×100 (%) |
|---|---|---|---|
| 1 | 4.78 | 0.23 | 4.81 |
| 2 | 6.47 | 0.36 | 5.56 |
| 3 | 5.88 | 0.34 | 5.78 |
| 4 | 11.86 | 0.36 | 3.04 |
| 5 | 18.67 | 1.18 | 6.32 |

arrive at the trunk, or the tree trunk is surrounded by obstacles, such as grasses, tree branches, and leaves, where the trunk points cannot be distinguished from the other surrounding points

$$Ht = Z_1 - Z_n \tag{13}$$

$$Fd = 2 \times \max\left(\mathrm{Dis}_{pg}\right) \tag{14}$$

where $Z_1$ and $Z_n$ are the first and the last values in matrix $Z$ (4). $\mathrm{Dis}_{pg}$ is the distance between the gravity center $g$ and the given point (11).

In the case of the presence of distinguishable trunk LiDAR points (trees number 1–5 in Fig. 5), the trunk can be modeled and measured. For this purpose, the middle point of the bounding box (1) must be located at the trunk center, supposing that the trunk has a cylindrical geometric form. Unfortunately, in most cases, the middle point of the bounding box is not congruent with the trunk center, which is why only one side of the trunk appears in the tree models (Trees 1b–5b in Fig. 5). Moreover, if the middle point of the bounding box is congruent with the trunk center, but the trunk is not covered sufficiently by LiDAR points, only the parts of the trunk covered by LiDAR points will also appear in the tree model. To solve this issue, each matrix of (2)–(4) must be divided into two submatrices: the upper submatrix that models the tree crown, and the bottom submatrix that models the tree trunk. Every one of these two submatrices must be calculated independently, and then the couples of submatrices must be merged to get the complete tree matrix. To calculate the trunk matrix, after descending sorting of tree point cloud according to the $Z$ coordinate values (see Section IV), the last four points of the tree point cloud are assigned to the trunk section; then, the mean of the selected points is calculated. Thereafter, the distance between the selected points and the mean is calculated. If the greatest distance is bigger than a given threshold, then this operation must stop, else the next upper point is added to the selected trunk points and the same procedure is repeated. This operation allows for detecting all trunk points [see Fig. 6(a)]. To determine the value of the used threshold, it can be noticed from Table III that the trunk radius of most trees is smaller than 10% of tree footprint radius. Consequently, a dynamic threshold value is considered to be 10% of tree footprint radius.

Once the tree point cloud is divided into crown and trunk clouds, the crown cloud can be modeled using the same approach, as suggested in Section IV [see Fig. 6(d)]. Concerning the trunk cloud, regarding the continuity of the trunk geometric

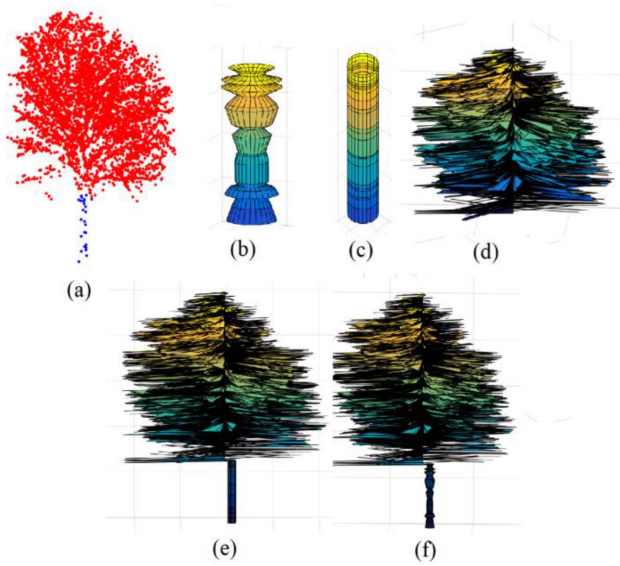


Fig. 6. Tree trunk modeling (Tree 4 in Fig. 5). (a) Division of the tree point cloud into crown and trunk clouds. (b) Trunk model considering the points covering the trunk. (c) Trunk model considering constant radius. (d) Crown model. (e) Tree model with constant trunk radius. (f) Tree model with nonconstant trunk radius.

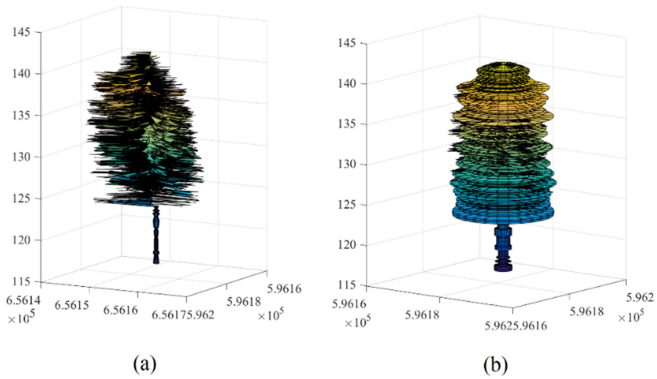


Fig. 7. Tree 4 in Fig. 5. (a) Tree model. (b) Adopted tree model for calculating the CV.

form and the popularity of vertical trunks, the trunk point cloud is suggested to be modeled using the same algorithm proposed by Tarsha Kurdi et al. [72], which is developed to model the multirotunda buildings [see Fig. 6(b)]. Another choice can be shown in Fig. 6(c), which supposes that the trunk can be modeled as a cylinder. However, the second choice is more suitable because the tree trunk has a normally chaotic asymmetrical geometric form. Figs. 6(e) and 7(f) illustrate tree number 4 in Fig. 5 with variant and constant diameter trunk models consecutively. To consider a more general case where the trunk is not always vertical and may have a variant diameter, the approach suggested by Xu et al. [61] can be adapted in future studies to extract the tree skeleton and then the skeleton point cloud will be divided into small horizontal slices and then a submatrix will be calculated for each slice. Thereafter, all trunk skeleton submatrices will be merged to calculate the complete skeleton submatrix, which can further be merged with the tree crown submatrix to calculate the whole tree matrix.

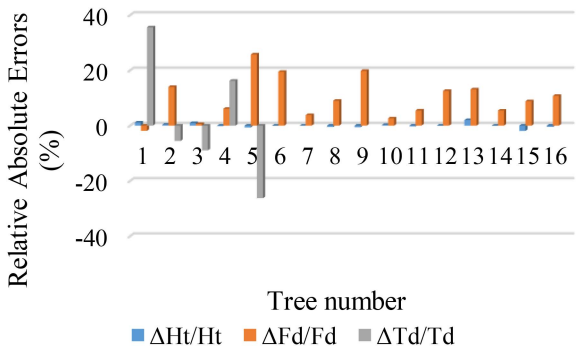


Fig. 8. RAEs of the calculated tree parameters. *Ht*: tree height; *Fd*: tree footprint diameter; and *Td*: trunk diameter.

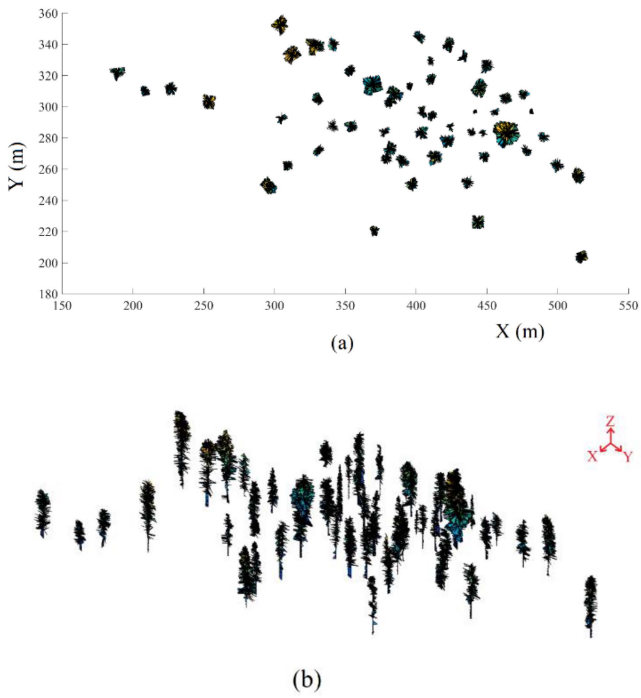


Fig. 9. Visualization of complete vegetation matrix. (a) 2-D visualization. (b) 3-D visualization.

At this stage, it is important to discuss the calculation of the tree's AGB. According to Herrero-Huerta et al. [67], the tree's AGB consists of the trunk and the crown. To calculate the canopy volume (CV) or the base crown volume (BCV), which represents the trunk volume, the air spaces inside the crown and between leaves and branches will be considered full. Hence, this article proposes to construct the tree rotating-surface model suggested by Tarsha Kurdi et al. [72], which simplifies the question of CV calculation. Fig. 7 shows two tree models, where Fig. 7(a) represents the model constructed by the suggested approach in this article, whereas Fig. 7(b) shows the tree model that will be used to calculate the AGB. Unfortunately, the trunk diameter in Fig. 9(b) is greater than the truth because it considers the tree footprint gravity center instead of the trunk footprint gravity center. Furthermore, Fig. 7(b) model supposes that the tree is symmetrical, which is not the case. Nevertheless, the tree BCV equals the volume summation of the frustum of cones forming the tree model. More investigations are envisaged in future effort to reduce the committed errors of BCV calculation.

Table IV further illustrates a numerical comparison between the tree parameters of direct measurement on the point clouds using the Cloudcompare software and the automatically extracted parameters from the constructed tree models. This comparison concerns the trees, as shown in Fig. 5. Table IV also mentions the AGB volume values (BCV+CV). The values of trunk diameters (*Td*) are calculated only for the trees having distinguishable trunk points (Trees 1–5). Furthermore, Table IV and Fig. 8 illustrate the relative absolute errors (RAEs) of extracted tree parameters in the same table. On the one hand, it can be noted that the tree heights are accurately estimated regarding the constructed tree models where max ($\Delta Ht$) = 21 cm and max RAE $Ht$ = 2.01%. On the other hand, the tree footprint and trunk diameters sometimes hold considerable errors (e.g., $\Delta Fd = 4.81$ m and RAE $F = 25.76\%$). In fact, this error can be explained through (14), which considers the crown and trunk to have a circular geometric form that envelopes the tree points. That is why, it multiplies the radius value by 2 to calculate the circle diameter value. The same issue can be met when the trunk diameter value is estimated. Indeed, the irregularity of the geometric form of the tree trunk can sometimes cause a considerable difference between the direct and automatic diameter estimation, e.g., in Tree 5, $\Delta Td = 0.31$ m.

However, in spite of the differences in tree parameter values between the direct measurement and the automatic calculation from the constructed tree model, the automatically extracted parameters are still acceptable regarding the adopted hypothesis of the tree's geometrical form.

At this stage, it is shown that the modeling method is beneficial to visualize all modeled trees of the same scanned scene. Below we further demonstrate that a tree can be expressed mathematically by three matrices *X*, *Y*, and *Z*. For one scanned scene with a $q$ number of trees, whole trees can be expressed within three matrices $X_v$, $Y_v$, and $Z_v$ (15)–(17)

$$X_v = \begin{bmatrix} X_1 & X_2 & \dots & X_q \end{bmatrix} \tag{15}$$

$$Y_v = \begin{bmatrix} Y_1 & Y_2 & \dots & Y_q \end{bmatrix} \tag{16}$$

$$Z_v = \begin{bmatrix} Z_1 & Z_2 & \dots & Z_q \end{bmatrix} \tag{17}$$

where $X_i$, $Y_i$, and $Z_i$ ($i = 1$ to $q$) are the three matrices that represent one tree, and $q$ is the number of trees in the scanned scene.

To visualize the complete vegetation matrix, a loop from 1 to $q$ can be used to read and visualize the individual tree matrices. Fig. 9 shows the 2-D and 3-D visualization of the complete scanned scene of 55 trees. From Fig. 9, it can be noted that the target trees are of different species and have different volumes, heights, and crown diameters. At this stage, the illustrated results in Fig. 9 confirm the effectiveness of the suggested tree modeling approach.

TABLE IV
TREE PARAMETERS OF DIRECT MEASUREMENT AND AUTOMATICALLY EXTRACTED FROM THE MODEL

| Tree Num | Species | Automatic extracted from tree Model | | | | Direct measurement on tree point cloud | | | Absolute errors | | | RAEs | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BCV+CV ($m^3$) | Ht (m) | Fd (m) | Td (m) | Ht (m) | Fd (m) | Td (m) | ΔHt (m) | ΔFd (m) | ΔTd (m) | ΔHt/Ht ×100 (%) | ΔFd/Fd ×100 (%) | ΔTd/Td ×100 (%) |
| 1 | Birch | 268.61 | 21.72 | 9.37 | 0.61 | 21.48 | 9.55 | 0.45 | 0.24 | 0.18 | 0.16 | 1.12 | 1.88 | 35.56 |
| 2 | Maple | 541.58 | 21.3 | 14.75 | 0.68 | 21.22 | 12.94 | 0.72 | 0.08 | 1.81 | 0.04 | 0.38 | 13.99 | 5.56 |
| 3 | Maple | 336.01 | 18.43 | 11.83 | 0.61 | 18.26 | 11.76 | 0.67 | 0.17 | 0.07 | 0.06 | 0.93 | 0.6 | 8.96 |
| 4 | Beech | 1273.58 | 25.45 | 25.15 | 0.86 | 25.5 | 23.71 | 0.74 | 0.05 | 1.44 | 0.12 | 0.2 | 6.07 | 16.22 |
| 5 | Oak | 1755.86 | 26.2 | 23.48 | 0.87 | 26.38 | 18.67 | 1.18 | 0.18 | 4.81 | 0.31 | 0.68 | 25.76 | 26.27 |
| 6 | Maple | 995.47 | 22.9 | 19.7 | - | 22.91 | 16.49 | - | 0.01 | 3.21 | - | 0.04 | 19.47 | - |
| 7 | Birch | 263 | 18.78 | 9.29 | - | 18.78 | 8.95 | - | 0 | 0.34 | - | 0 | 3.8 | - |
| 8 | Birch | 508.32 | 17.81 | 14.77 | - | 17.9 | 13.55 | - | 0.09 | 1.22 | - | 0.5 | 9 | - |
| 9 | Maple | 214.02 | 12.12 | 10.76 | - | 12.19 | 8.98 | - | 0.07 | 1.78 | - | 0.57 | 19.82 | - |
| 10 | Beech | 1755.4 | 23.98 | 21.24 | - | 23.92 | 20.7 | - | 0.06 | 0.54 | - | 0.25 | 2.61 | - |
| 11 | Maple | 467.72 | 14.59 | 15.45 | - | 14.63 | 14.65 | - | 0.04 | 0.8 | - | 0.27 | 5.46 | - |
| 12 | Walnut | 769.73 | 18.76 | 17.92 | - | 18.77 | 15.92 | - | 0.01 | 2 | - | 0.05 | 12.56 | - |
| 13 | Fir | 100.62 | 10.64 | 6.21 | - | 10.43 | 5.49 | - | 0.21 | 0.72 | - | 2.01 | 13.11 | - |
| 14 | Maple | 489.9 | 14.46 | 14.38 | - | 14.47 | 13.64 | - | 0.01 | 0.74 | - | 0.07 | 5.43 | - |
| 15 | Pine | 89.9 | 7.64 | 7.79 | - | 7.79 | 7.16 | - | 0.15 | 0.63 | - | 1.93 | 8.8 | - |
| 16 | Maple | 701.2 | 19.66 | 17.19 | - | 19.75 | 15.52 | - | 0.09 | 1.67 | - | 0.46 | 10.76 | - |

BCV: Base crown volume; CV: canopy volume; Ht: tree height; Fd: tree footprint diameter; and Td: trunk diameter.
Tree IDs are taken from Fig. 5.

### D. Comparison With a Similar Algorithm

It is necessary, at this stage, to highlight the differences and originality aspects of the presented algorithm regarding the building modeling algorithm suggested by Tarsha Kurdi et al. [72]. Indeed, the two algorithms are based on the hypothesis of rotating surfaces, and they use three matrices to visualize the calculated models; however, the focus is totally different. The present approach targets the trees class, whereas the algorithm suggested by Tarsha Kurdi et al. [72] targets buildings. To be specific, they use three matrices *X, Y*, and *Z* to express and visualize the constructed model. The input data are the LiDAR point cloud of the target object.

On the other hand, the building modeling algorithm calculates the vertical cross section to construct the basic building model and keeps being improved by considering all building LiDAR points. However, the present algorithm calculates the tree model without using the tree's vertical cross section. The present algorithm does not need to be improved, such as the building modeling algorithm, because it considers all points of the LiDAR point cloud. Furthermore, the present algorithm considers the discontinuity geometrical nature of the tree crown, whereas the building modeling algorithm considers the geometrical continuity nature of the buildings. The tree modeling algorithm distinguishes between the crown geometry and the truck geometry, but the building modeling algorithm does not distinguish between different parts of the building.

To conclude, despite the same mathematical principle being used in the last two modeling algorithms, each algorithm is adapted to be harmonized with the different geometrical nature of the target objects in the two cases.

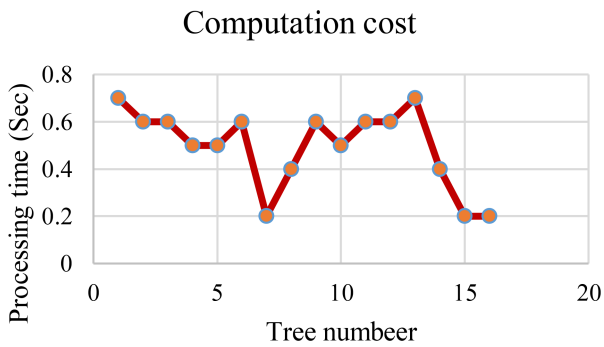


Fig. 10. Computation cost of the suggested tree modeling algorithm for trees, as mentioned in Table IV.

### E. Computation Cost

It is important to underline the computational cost of the suggested modeling approach. In this article, we used an Intel(R) Core (TM) i7-10610U CPU @ 1.80 GHz 2.30 GHz, RAM 32.0 GB (31.6 GB usable). Moreover, the code is developed and tested under MATLAB 2021b.

Fig. 10 shows the computation cost, as mentioned in Table IV. It can be noted that regarding the modest used computer quality, the processing time of the tested tree point-cloud samples is reasonable where the average processing time is about 0.5 s by tree. Such low computational cost may be explained by the simplicity of the modeling suggested algorithm regarding the huge number of LiDAR points, which is considered a great gain in the context of 3-D tree modeling from LiDAR data and would be beneficial for large-scale forest inventory at the level of individual trees, especially considering that the performance can

be further improved with more efficient implementation, such as with Python.

## VI. Conclusion

There is a need for automatic mass modeling of individual trees. This article proposed to build them on the basis of the segmented LiDAR points of single trees. It assumes that the tree trunk is vertical and can be modeled by multiple layers depending on the height of the trees. Under these assumptions, the tree model is represented by three matrices. The assumed number of height levels corresponds to the number of rows of the model matrix, while the point density contributes to the determination of the number of columns of the model matrix. The adopted number of rows and columns of the matrix determines the structure of the model and enables a 3-D visualization of the tree. Thus, the accuracy of the model may be variable. This assumption simplifies the model and allows it to be made quickly. Despite this simplification, thanks to the proposed algorithm, the shape of the crown corresponds to reality with the calculated accuracy. The point density, the distance between the point and the gravity center, and the relation of one point to the other play an essential role in the accuracy of tree modeling. By using the proposed model, the mean accuracy achieved for the 55 tested trees is between 0.3 and 0.89 m, while the relative accuracy is between 04% and 17.5%. The released tests demonstrated that this development is a simple and effective algorithm for 3-D modeling of individual trees and can be used for tree parameter calculation.

The proposed method of generating 3-D trees differs significantly from those presented in Section II. Although many tree models in the literature can consider the tree trunk if the trunk LiDAR points are distinguishable, they do not consider the complete skeleton of the tree and are unable to form one coherent crown body, causing a high generalization that leads to a significant deformation of the trees. Instead, our model is physical volumetric models of the space occupied by the tree. It generates a layered-segmented structure, reflecting a more realistic shape of the tree. The algorithm is simpler and can be potentially used for mass production and adopted as a general tree modeling approach in the OGC CityGML standard.

More investigations are needed in future work to consider the general trunk geometry and integrate the tree skeleton within the constructed model. Moreover, the tree solid model will be used to calculate the AGB volume.

## Acknowledgment

The authors would like to thank P. Reed, the Managing Director of East Coast Surveys (Aust.) Pty, Ltd., and ScanX-Plus Pty, Ltd., companies for supporting this work, http://www.eastcoastsurveys.com.au (Accessed on: September 5, 2023).

## References

[1] F. Tarsha Kurdi, Z. Gharineiat, G. Campbell, M. Awrangjeb, and E. K. Dey, "Automatic filtering of LiDAR building point cloud in case of trees associated to building roof," *Remote Sens.*, vol. 14, 2022, Art. no. 430, doi: 10.3390/rs14020430.

[2] K. Calders et al., "Large-area virtual forests from terrestrial laser scanning data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 1765–1767, doi: 10.1109/IGARSS.2016.7729452.

[3] J. Wang and K. Xu, "Shape detection from raw LiDAR data with subspace modeling," *IEEE Trans. Visual. Comput. Graph.*, vol. 23, no. 9, pp. 2137–2150, Sep. 2017, doi: 10.1109/TVCG.2016.2601915.

[4] F. Tarsha Kurdi, P. Reed, Z. Gharineiat, and M. Awrangjeb, "Efficiency of terrestrial laser scanning in survey works: Assessment, modelling, and monitoring," *Int. J. Environ. Sci. Natural Resour.*, vol. 32, no. 2, 2023, Art. no. 556334, doi: 10.19080/IJESNR.2023.32.556334.

[5] G. Schwoch, T. J. Lieb, and H. Lenz, "Flight testing drone contingencies during runway inspection in U-space shared airspace," in *Proc. Integr. Commun., Navig. Surveill. Conf.*, 2023, pp. 1–7, doi: 10.1109/ICNS58246.2023.10124322.

[6] S. J. McTegg, F. Tarsha Kurdi, S. Simmons, and Z. Gharineiat, "Comparative approach of unmanned aerial vehicle restrictions in controlled airspaces," *Remote Sens.*, vol. 14, 2022, Art. no. 822, doi: 10.3390/rs14040822.

[7] X. Zhou and W. Li, "A geographic object-based approach for land classification using LiDAR elevation and intensity," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 669–673, May 2017, doi: 10.1109/LGRS.2017.2669994.

[8] E. Bratsolis, S. Gyftakis, E. Charou, and N. Vassilas, "Comparative analysis of classification techniques for building block extraction using aerial imagery and LiDAR data," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, 2013, pp. 000080–000085, doi: 10.1109/ISSPIT.2013.6781858.

[9] D. Marinelli, C. Paris, and L. Bruzzone, "An approach based on deep learning for tree species classification in LiDAR data acquired in mixed forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jun. 2022, Art. no. 7004305, doi: 10.1109/LGRS.2022.3181680.

[10] F. Tarsha Kurdi, M. Awrangjeb, and N. Munir, "Automatic filtering and 2D modeling of airborne laser scanning building point cloud," *Trans. GIS*, vol. 25, no. 1, pp. 164–188, Feb. 2021, doi: 10.1111/tgis.12685.

[11] Z. Gharineiat, F. Tarsha Kurdi, and G. Campbell, "Review of automatic processing of topography and surface feature identification LiDAR data using machine learning techniques," *Remote Sens.*, vol. 14, no. 19, 2022, Art. no. 4685, doi: 10.3390/rs14194685.

[12] E. K. Dey, M. Awrangjeb, F. Tarsha Kurdi, and B. Stantic, "Machine learning-based segmentation of aerial LiDAR point cloud data on building roof," *Eur. J. Remote Sens.*, vol. 56, no. 1, 2023, Art. no. 2210745, doi: 10.1080/22797254.2023.2210745.

[13] J. Summit and R. Sommer, "Further studies of preferred tree shapes," *Environ. Behav.*, vol. 31, pp. 550–576, 1999, doi: 10.1177/00139169921972236.

[14] CityGML - Open Geospatial Consortium, Accessed on: Apr. 19, 2023. [Online]. Available: https://www.ogc.org/standard/citygml/

[15] X. Wang, Y.-P. Luo, T. Jiang, H. Gong, S. Luo, and X.-W. Zhang, "A new classification method for LiDAR data based on unbalanced support vector machine," in *Proc. Int. Symp. Image Data Fusion*, 2011, pp. 1–4, doi: 10.1109/ISIDF.2011.6024312.

[16] F. Tarsha Kurdi, W. Amakhchan, and Z. Gharineiat, "Random forest machine learning technique for automatic vegetation detection and modelling in LiDAR data," *Int. J. Environ. Sci. Natural Resour.*, vol. 28, no. 2, 2021, Art. no. 556234, doi: 10.19080/IJESNR.2021.28.556234.

[17] C. Beil, R. Ruhdorfer, T. Coduro, and T. H. Kolbe, "Detailed streetspace modelling for multiple applications: Discussions on the proposed CityGML 3.0 transportation model," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 10, 2020, Art. no. 603, doi: 10.3390/IJGI9100603.

[18] F. Tarsha Kurdi and M. Awrangjeb, "Comparison of LiDAR building point cloud with reference model for deep comprehension of cloud structure," *Can. J. Remote Sens.*, vol. 46, no. 5, pp. 603–621, 2020, doi: 10.1080/07038992.2020.1829462.

[19] T. Kutzner, K. Chaturvedi, and T. H. Kolbe, "CityGML 3.0: New functions open up new applications," *PFG –J. Photogramm., Remote Sens. Geoinf. Sci.*, vol. 88, no. 1, pp. 43–61, 2020, doi: 10.1007/S41064-020-00095-Z.

[20] K. Chaturvedi, A. Matheus, S. H. Nguyen, and T. H. Kolbe, "Securing spatial data infrastructures for distributed smart city applications and services," *Future Gener. Comput. Syst.*, vol. 101, pp. 723–736, 2019, doi: 10.1016/J.FUTURE.2019.07.002.

[21] C. Beil and T. H. Kolbe, "CITYGML and the streets of New York—A proposal for detailed street space modelling," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. IV-4/W5, pp. 9–16, 2017, doi: 10.5194/ISPRS-ANNALS-IV-4-W5-9-2017.

[22] B. Schwab, C. Beil, and T. H. Kolbe, "Spatio-semantic road space modeling for vehicle–pedestrian simulation to test automated driving systems," *Sustainability*, vol. 12, no. 9, 2020, Art. no. 3799, doi: 10.3390/SU12093799.

[23] J. P. Richa, J.-E. Deschaud, F. Goulette, and N. Dalmasso, "AdaSplats: Adaptive splatting of point clouds for accurate 3D modeling and real-time high-fidelity lidar simulation," *Remote Sens.*, vol. 14, no. 24, 2022, Art. no. 6262, doi: 10.3390/RS14246262.

[24] C. Plachetka, J. Rieken, and M. Maurer, "The TUBS road user dataset: A new LiDAR dataset and its application to CNN-based road user classification for automated vehicles," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2623–2630, doi: 10.1109/ITSC.2018.8569765.

[25] P. Jayaraj and A. M. Ramiya, "3D CITYGML building modelling from LiDAR point cloud data," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-5, no. 5, pp. 175–180, 2018, doi: 10.5194/ISPRS-ARCHIVES-XLII-5-175-2018.

[26] Z. Zhang et al., "Vectorized rooftop area data for 90 cities in China," *Sci. Data*, vol. 9, no. 1, 2022, Art. no. 66, doi: 10.1038/s41597-022-01168-x.

[27] R. Peters, B. Dukai, S. Vitalis, J. van Liempt, and J. Stoter, "Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of The Netherlands," *Photogramm. Eng. Remote Sens.*, vol. 88, no. 3, pp. 165–170, 2022, doi: 10.14358/PERS.21-00032R2.

[28] N. Milojevic-Dupont et al., "EUBUCCO v0.1: European building stock characteristics in a common and open database for 200+ million individual buildings," *Sci. Data*, vol. 10, 2023, Art. no. 147, doi: 10.1038/s41597-023-02040-2.

[29] E. Lewandowicz, F. Tarsha Kurdi, and Z. Gharineiat, "3D LoD2 and LoD3 modeling of buildings with ornamental towers and turrets based on LiDAR data," *Remote Sens.*, vol. 14, no. 19, 2022, Art. no. 4687, doi: 10.3390/RS14194687.

[30] L. Xiong et al., "Quantifying mangrove canopy regrowth and recovery after Hurricane Irma with large-scale repeat airborne LiDAR in the Florida Everglades," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 114, 2022, Art. no. 103031, doi: 10.1016/J.JAG.2022.103031.

[31] A. Adhikari, C. R. Montes, and A. Peduzzi, "A comparison of modeling methods for predicting forest attributes using LiDAR metrics," *Remote Sens.*, vol. 15, no. 5, 2023, Art. no. 1284, doi: 10.3390/rs15051284.

[32] L. Lombard, R. Ismail, and N. K. Poona, "Modelling forest species using LiDAR-derived metrics of forest canopy gaps," *South Afr. J. Geomatics*, vol. 9, no. 1, pp. 31–43, 2022, doi: 10.4314/sajg.v9i1.3.

[33] J. Hyyppä, H. Hyyppä, D. Leckie, F. Gougeon, X. Yu, and M. Maltamo, "Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests," *Int. J. Remote Sens.*, vol. 29, no. 5, pp. 1339–1366, 2008, doi: 10.1080/01431160701736489.

[34] Z. Zhen, L. J. Quackenbush, and L. Zhang, "Trends in automatic individual tree crown detection and delineation—Evolution of LiDAR data," *Remote Sens.*, vol. 8, no. 4, 2016, Art. no. 333, doi: 10.3390/RS8040333.

[35] X. Peng et al., "Comparison of modeling algorithms for forest canopy structures based on UAV-LiDAR: A case study in tropical China," *Forests*, vol. 11, no. 12, 2020, Art. no. 1324, doi: 10.3390/f11121324.

[36] J. Lei, H. Li, S. Zhao, Y. Wang, Y. Jiang, and G. Zhu, "Automatic identification of street trees with improved RandLA-Net and accurate calculation of shading area with density-based iterative $\alpha$-shape," *IEEE Access*, vol. 10, pp. 132384–132395, 2022, doi: 10.1109/ACCESS.2022.3229901.

[37] G. Yue, R. Liu, H. Zhang, and M. Zhou, "A method for extracting street trees from mobile LiDAR point clouds," *Open Cybern. Systemics J.*, vol. 9, no. 1, pp. 204–209, 2015, doi: 10.2174/1874110X01509010204.

[38] Y. Wang, Y. Lin, H. Cai, and S. Li, "Hierarchical fine extraction method of street tree information from mobile LiDAR point cloud data," *Appl. Sci.*, vol. 13, no. 1, 2023, Art. no. 276, doi: 10.3390/APP13010276.

[39] S. Pu, M. Rutzinger, G. Vosselman, and S. O. Elberink, "Recognizing basic structures from mobile laser scanning data for road inventory studies," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 6, pp. S28–S39, Dec. 2011, doi: 10.1016/J.ISPRSJPRS.2011.08.006.

[40] R. B. Reckziegel, E. Larysch, J. P. Sheppard, H.-P. Kahle, and C. Morhart, "Modelling and comparing shading effects of 3D tree structures with virtual leaves," *Remote Sens.*, vol. 13, no. 3, 2021, Art. no. 532, doi: 10.3390/RS13030532.

[41] Eure Lex, Regulation (EU) 2018/841 of the European Parliament and of the Council of 30 May 2018 on the Inclusion of Greenhouse Gas Emissions and Removals From Land Use, Land Use Change and Forestry in the 2030 Climate and Energy Framework, and Amending Regulation (EU) No 525/2013 and Decision No 529/2013/EU (Text With EEA Relevance), 2018. [Online]. Available: http://data.europa.eu/eli/reg/2018/841/oj

[42] M. P. McClelland, J. van Aardt, and D. Hale, "Manned aircraft versus small unmanned aerial system—Forestry remote sensing comparison utilizing LiDAR and structure-from-motion for forest carbon modeling and disturbance detection," *J. Appl. Remote Sens.*, vol. 14, no. 2, 2020, Art. no. 022202, doi: 10.1117/1.JRS.14.022202.

[43] F. Jiang, H. Sun, K. Ma, L. Fu, and J. Tang, "Improving aboveground biomass estimation of natural forests on the Tibetan Plateau using spaceborne LiDAR and machine learning algorithms," *Ecol. Indicators*, vol. 143, 2022, Art. no. 109365, doi: 10.1016/J.ECOLIND.2022.109365.

[44] M. Liang, L. Duncanson, J. A. Silva, and F. Sedano, "Quantifying aboveground biomass dynamics from charcoal degradation in Mozambique using GEDI LiDAR and Landsat," *Remote Sens. Environ.*, vol. 284, 2022, Art. no. 113367, doi: 10.1016/J.RSE.2022.113367.

[45] D. Michelini et al., "Hyperspectral and LiDAR data for the prediction via machine learning of tree species, volume and biomass: A contribution for updating forest management plans," in *Proc. Italian Conf. Geomatics Geospatial Technol.*, 2022, vol. 1651, pp. 235–250, doi: 10.1007/978-3-031-17439-1_17.

[46] Z. A. Pierrat et al., "Forests for forests: Combining vegetation indices with solar-induced chlorophyll fluorescence in random forest models improves gross primary productivity prediction in the boreal forest," *Environ. Res. Lett.*, vol. 17, no. 12, 2022, Art. no. 125006, doi: 10.1088/1748-9326/ACA5A0.

[47] L. Bauer, N. Knapp, and R. Fischer, "Mapping Amazon forest productivity by fusing GEDI LiDAR waveforms with an individual-based forest model," *Remote Sens.*, vol. 13, no. 22, 2021, Art. no. 4540, doi: 10.3390/RS13224540.

[48] E. Rödig, A. Huth, F. Bohn, C. Rebmann, and M. Cuntz, "Estimating the carbon fluxes of forests with an individual-based forest model," *Forest Ecosyst.*, vol. 4, no. 1, 2017, Art. no. 4, doi: 10.1186/S40663-017-0091-1.

[49] A. Glad, B. Reineking, M. Montadert, A. Depraz, and J.-M. Monnet, "Assessing the performance of object-oriented LiDAR predictors for forest bird habitat suitability modeling," *Remote Sens. Ecol. Conservation*, vol. 6, no. 1, pp. 5–19, 2020, doi: 10.1002/RSE2.117.

[50] S. Xu, N. Ye, S. Xu, and F. Zhu, "A supervoxel approach to the segmentation of individual trees from LiDAR point clouds," *Remote Sens. Lett.*, vol. 9, no. 6, pp. 515–523, 2018, doi: 10.1080/2150704X.2018.1444286.

[51] M. Weinmann, M. Weinmann, C. Mallet, and M. Bredif, "A classification-segmentation framework for the detection of individual trees in dense MMS point cloud data acquired in urban areas," *Remote Sens.*, vol. 9, no. 3, 2017, Art. no. 277, doi: 10.3390/RS9030277.

[52] F. J. Fischer, I. Maréchaux, and J. Chave, "Improving plant allometry by fusing forest models and remote sensing," *New Phytologist*, vol. 223, no. 3, pp. 1159–1165, 2019, doi: 10.1111/nph.15810.

[53] Z. Huang et al., "Retrieval of aerodynamic parameters in rubber tree forests based on the computer simulation technique and terrestrial laser scanning data," *Remote Sens.*, vol. 12, no. 8, 2020, Art. no. 1318, doi: 10.3390/RS12081318.

[54] Z. Hui, S. Jin, Y. Xia, L. Wang, Y. Y. Ziggah, and P. Cheng, "Wood and leaf separation from terrestrial LiDAR point clouds based on mode points evolution," *ISPRS J. Photogramm. Remote Sens.*, vol. 178, pp. 219–239, 2021, doi: 10.1016/J.ISPRSJPRS.2021.06.012.

[55] C. Zhu, X. Zhang, B. Hu, and M. Jaeger, "Reconstruction of tree crown shape from scanned data," in *Proc. Int. Conf. Technol. E-Learning Digital Entertainment*, 2008, pp. 745–756, doi: 10.1007/978-3-540-69736-7_79.

[56] M. Béland, J.-L. Widlowski, R. A. Fournier, J.-F. Côté, and M. M. Verstraete, "Estimating leaf area distribution in savanna trees from terrestrial LiDAR measurements," *Agricultural Forest Meteorol.*, vol. 151, no. 9, pp. 1252–1266, 2011, doi: 10.1016/J.AGRFORMET.2011.05.004.

[57] S. A. Ahmadi, A. Ghorbanian, F. Golparvar, A. Mohammadzadeh, and S. Jamali, "Individual tree detection from unmanned aerial vehicle (UAV) derived point cloud data in a mixed broadleaf forest using hierarchical graph approach," *Eur. J. Remote Sens.*, vol. 55, no. 1, pp. 520–539, 2022, doi: 10.1080/22797254.2022.2129095.

[58] Z. Wang et al., "A structure-aware global optimization method for reconstructing 3-D tree models from terrestrial laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5653–5669, Sep. 2014, doi: 10.1109/TGRS.2013.2291815.

[59] S. M. Krishna Moorth, K. Calders, M. B. Vicari, and H. Verbeeck, "Improved supervised learning-based approach for leaf and wood classification from LiDAR point clouds of forests," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3057–3070, May 2020, doi: 10.1109/TGRS.2019.2947198.

[60] W. Zhu, C. Zhu, and Y. Zhang, "Research on deep learning individual tree segmentation method coupling RetinaNet and point cloud clustering," *IEEE Access*, vol. 9, pp. 126635–126645, 2021, doi: 10.1109/ACCESS.2021.3111654.
[61] J. Xu, J. Shan, and G. Wang, "Hierarchical modeling of street trees using mobile laser scanning," *Remote Sens.*, vol. 12, no. 14, 2020, Art. no. 2321, doi: 10.3390/RS12142321.
[62] J. Mei, L. Zhang, S. Wu, Z. Wang, and L. Zhang, "3D tree modeling from incomplete point clouds via optimization and $L_1$-MST," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 5, pp. 999–1021, 2017, doi: 10.1080/13658816.2016.1264075.
[63] A. Bucksch and R. Lindenbergh, "CAMPINO—A skeletonization method for point cloud processing," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 1, pp. 115–127, 2008, doi: 10.1016/J.ISPRSJPRS.2007.10.004.
[64] A. Bucksch, R. Lindenbergh, and M. Menenti, "SkelTre: Robust skeleton extraction from imperfect point clouds," *Vis. Comput.*, vol. 26, no. 10, pp. 1283–1300, 2012, doi: 10.1007/S00371-010-0520-4.
[65] S. Du, R. Lindenbergh, H. Ledoux, J. Stoter, and L. Nan, "AdTree: Accurate, detailed, and automatic modelling of laser-scanned trees," *Remote Sens.*, vol. 11, no. 18, 2019, Art. no. 2074, doi: 10.3390/RS11182074.
[66] X. Liu et al., "A novel entropy-based method to quantify forest canopy structural complexity from multiplatform LiDAR point clouds," *Remote Sens. Environ.*, vol. 282, 2022, Art. no. 113280, doi: 10.1016/j.rse.2022.113280.
[67] M. Herrero-Huerta, R. Lindenbergh, and P. Rodriguez-Gonzâlvez, "Automatic tree parameter extraction by a mobile LiDAR system in an urban context," *PLoS One*, vol. 13, no. 4, 2018, Art. no. e0196004, doi: 10.1371/journal.pone.0196004.
[68] Q. Shu, T. Rötzer, A. Detter, and F. Ludwig, "Tree information modeling: A data exchange platform for tree design and management," *Forests*, vol. 13, no. 11, 2022, Art. no. 1955, doi: 10.3390/F13111955.
[69] L. M. Ortega-Córdova, "Urban vegetation modeling 3D levels of detail," 2018. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3A8b8967a8-0a0f-498f-9d37-71c6c3e532af
[70] M. Dai and G. Li, "Soft segmentation and reconstruction of tree crown from laser scanning data," *Electronics*, vol. 12, no. 10, 2023, Art. no. 2300, doi: 10.3390/electronics12102300.
[71] E. Lewandowicz and M. Antolak, "Converting database on dendrological objects of Kortowo campus at the University of Warmia and Mazury in Olsztyn to current GIS standards," in *Proc. 15th Int. Multidisciplinary Sci. GeoConf.*, 2015, vol. 2, pp. 787–794.
[72] F. Tarsha Kurdi, E. Lewandowicz, Z. Gharineiat, and J. Shan, "Modeling multi-rotunda buildings at LoD3 level from LiDAR data," *Remote Sens.*, vol. 15, no. 13, 2023, Art. no. 3324, doi: 10.3390/rs15133324.
[73] F. Tarsha Kurdi, T. Landes, P. Grussenmeyer, and M. Koehl, "Model-driven and data-driven approaches using LiDAR data: Analysis and comparison," in *Proc. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, 2006, pp. 87–92.

**Fayez Tarsha Kurdi** received the Ph.D. degree in geomatics and automatic processing of LiDAR data from the Graduate School of Science and Technology of Strasbourg (INSA of Strasbourg), ULP University (Strasbourg I), Strasbourg, France, in 2008, with ten years of industrial experience in surveying and laser scanning.

He received various positive appraisals. He was a Fellow Researcher with the Institute for Integrated and Intelligent Systems, Griffith University, Australia, in 2019. He has been an Adjunct Lecturer and a Research Assistant with the School of Civil Engineering and Surveying, University of Southern Queensland (UniSQ), Springfield, QLD, Australia, since 2021, with machine learning techniques and 3-D modeling being his current research focus. Keeping his position with UniSQ, he has worked with East Coast Surveys, Ltd., firm as a Survey Systems and Research Manager.

Dr. Tarsha Kurdi serves on the editorial boards of several remote sensing journals.

**Elżbieta Lewandowicz** received the habilitation degree in geoinformatics from the University of Warmia and Mazury in Olsztyn, Olsztyn, Poland, in 2007.

She is currently a Professor with the Department of Geoinformation and Cartography, University of Warmia and Mazury in Olsztyn. She has authored or coauthored more than 100 scientific papers. Her research interests and teaching duties encompass areas related to GIS. She is interested in spatial data modeling and seeks algorithms for processing data to construct the skeletons of spatial objects. The construction of the digital twin of the world has prompted her interest in the three-dimensional modeling of spatial objects based on LiDAR data. In her leisure time, she is passionate about active kayaking tourism. She has combined her passion with scientific research, focusing on the assessment of geographic potential, infrastructure, and economic activities conducive to the development of active tourism while maintaining sustainable development. She is a member of the Polish Society for Spatial Information.

**Jie Shan** received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China.

He has worked with universities in China, Germany, and Sweden. He is currently a Professor with the School of Civil Engineering, and the Department of Earth, Atmospheric and Planetary Sciences, Purdue University, West Lafayette, IN, USA. His research interests include sensor geometry and positioning, object extraction and reconstruction from images and point clouds, urban remote sensing, pattern recognition, and geospatial data mining.

Dr. Shan was the recipient of multiple Best Paper Awards and elected ASPRS Fellow. He serves on the editorial boards of several remote sensing journals.

**Zahra Gharineiat** received the Ph.D. degree in surveying from the University of Newcastle, Callaghan, NSW, Australia, in 2012.

She currently holds the position of Associate Professor of surveying with the University of Southern Queensland (UniSQ), Springfield, QLD, Australia. In 2015, while pursuing her Ph.D. degree, she embarked on her academic career with UniSQ. During her academic career, she has developed a passion for unmanned aerial vehicles (UAVs) or drones and obtained my remote pilot licence. Her research interests and expertise include UAVs and their applications, light detection and ranging and automatic feature extraction, digital twins, applied machine learning, spatial data management, and 3-D visualisation, remote sensing, and climate modeling.