Modeling of the 3D Tree Skeleton Using Real-World Data: A Survey

José L. Cárdenas, Carlos J. Ogayar, Francisco R. Feito, and Juan M. Jurado

Abstract—Tree modeling has been extensively studied in computer graphics. Recent advances in the development of high-resolution sensors and data processing techniques are extremely useful for collecting 3D datasets of real-world trees and generating increasingly plausible branching structures. The wide availability of versatile acquisition platforms allows us to capture multi-view images and scanned data that can be used for guided 3D tree modeling. In this paper, we carry out a comprehensive review of the state-of-the-art methods for the 3D modeling of botanical tree geometry by taking input data from real scenarios. A wide range of studies has been proposed following different approaches. The most relevant contributions are summarized and classified into three categories: (1) procedural reconstruction, (2) geometry-based extraction, and (3) image-based modeling. In addition, we describe other approaches focused on the reconstruction process by adding additional features to achieve a realistic appearance of the tree models. Thus, we provide an overview of the most effective procedures to assist researchers in the photorealistic modeling of trees in geometry and appearance. The article concludes with remarks and trends for promising research opportunities in 3D tree modeling using real-world data.

Index Terms—Tree modeling, 3D reconstruction, real-world data processing, computational geometry, realistic rendering

1 Introduction

The geometric modeling of tree branching structures has been a long-standing problem in computer graphics. Vegetation plays an essential role in the realistic characterization of virtual scenes in urban and natural layouts. The generation of the 3D plant geometry is currently an open problem that has been studied by applying different techniques. In the last decades, a wide variety of methods have been proposed for the generation of 3D trees with a high level of realism. Additionally, the amount of captured 3D data is continuously increasing due to the rapid development of high-resolution cameras, scanners and multi-view stereo systems. The use of input data that partially represent the tree structure allows us to develop guided approaches to achieve a faithful modeling of its geometry and visual appearance. This study aims to provide an overview, classification and comparison of state-of-the-art methods developed over the years to model 3D trees from input data collected in the real world.

In the field of real-world data capturing, traditional technologies such as 3D laser scanning and photogrammetry have been enhanced by modern systems mounted on aerial platforms. The use of Unmanned Aerial Vehicles (UAVs) and additional sensors allows us to cover almost the entire tree structure. However, some branches cannot be modeled directly on the remotely sensed data, as they are mostly occluded by other overlapping branches and leaves. Despite this limitation, these data represent many visible structures that can be used as a useful guide for modeling tree geometry. To exploit the captured 3D data to extract tree geometry, a high-level representation is often used to generate the skeleton structure. In general, point clouds and simple geometric primitives like 3D polylines or voxels enable a simplification of the tree geometry.

The geometric modeling of trees is mostly focused on a virtual representation of botanical trees by a robust parametrization considering multiple geometric and spatial constraints. In this domain, the first study was presented by Ulam *et al.* [1] that created 3D tree structures based on cellular automata. On the other hand, a general overview was provided by Sen and Day *et al.* [2] that summarized the most prevalent contributions at the beginning of the research on tree modeling. These initial approaches were mainly based on the use of fractals and rule-based systems for modeling realistic tree structures. These methods range from the generation of trees only considering their shapes to a full-growth simulation according to their biological and physiological features. All these methods lack input from captured data, so to generate realistic trees conforming to the shape and characteristics of certain species, many parameters must be tweaked. Due to the tedious and complex parameterization of the procedural methods and the challenging modeling of some branches, the use of 3D input data of the tree structure involves many advantages that have been exploited in plant modeling.

- *José L. Cárdenas, Carlos J. Ogayar, and Francisco R. Feito are with the Department of Computer Science, University of Jaén, 23071 Jaén, Spain. E-mail: {jcdonoso, cogayar, ffeito}@ujaen.es.*
- *Juan M. Jurado is with the Department of Software Engineering, University of Granada, 18014 Granada, Spain. E-mail: jjurado@ugr.es.*

Manuscript received 4 October 2021; revised 18 July 2022; accepted 19 July 2022. Date of publication 21 July 2022; date of current version 10 November 2023.
This work was supported in part by MCIN/AEI/10.13039/501100011033, Junta de Andalucía (Spain), and European Union's ERDF funds under Projects PID2021-126339OB-I00, 1381202-GEU, and PYC20-RE-005-UJA, and in part by the Spanish Ministry of Science, Innovation and Universities through the Doctoral Grant to the first author under Grant FPU17/01902.
(Corresponding author: Juan M. Jurado.)
Recommended for acceptance by W. Wang.
Digital Object Identifier no. 10.1109/TVCG.2022.3193018

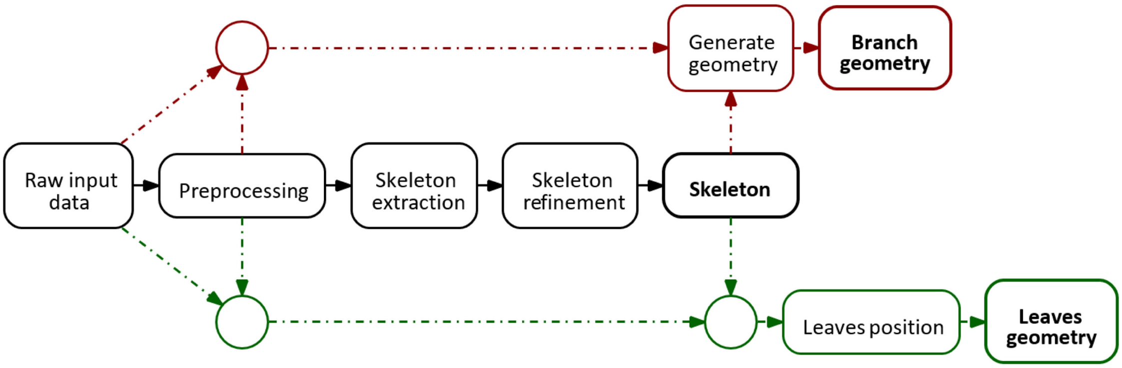


Fig. 1. General workflow for the 3D reconstruction of the tree geometry.

This comprehensive review describes the most relevant research focused on the generation of the 3D tree skeleton using real-world data. The tree skeleton is essential to simulate as accurately as possible the shape and morphology of natural entities. This is formed by the trunk and a complex branching ramification in the tree canopy. In addition to the tree skeleton, the appearance of these models can be enhanced with leaves, textures and other virtual assets [3], [4]. In computer graphics, modeling physically plausible trees has a long tradition that more recently also encompasses complex simulation of the tree growth. Indeed, this process is mainly determined by intrinsic features depending on the tree species, health status and other morphological and physiological traits. Moreover, the tree geometry may be affected by external factors such as incident light, proximity to other natural or artificial objects and meteorological effects. Previous studies only focused on a visual representation of the trees and do not offer enough information on how to model the changes of the tree geometry in a dynamic ecosystem [5]. An advantage of modeling the 3D skeleton is its ability to represent natural phenomena like the interaction of plants with the wind [6], [7], with the fire [8], [9], with fauna [10] as well as feedback loops between vegetation, soil, and the atmosphere at a local scale [11], [12]. The goal of many of these methods is to employ detailed geometric representations to faithfully simulate the underlying physical or biological processes. To this end, the generation of the 3D tree skeleton is a crucial task in plant modeling that has been addressed from multiple approaches.

In this study, our goal is to collect the existing methods that use real-world data as input to generate the tree geometry. Main post-processing techniques that aim to improve the tree's appearance are also described. In addition, a taxonomy of 3D tree modeling methods is proposed by classifying them into three main categories: (1) procedural reconstruction, (2) geometry-extraction algorithms, and (3) image-based modeling. Finally, a comparative analysis of the surveyed 3D tree reconstruction methods is presented according to the available input data.

This paper is organized as follows. In Section 2, we introduce the challenges and recent advances in tree reconstruction, including main contributions based on procedural modeling, geometry-based extraction and image-based modeling. Section 3 summarizes other methods focused on improving botanical tree structures by treating occlusion, transforming the tree skeleton into a 3D mesh and adding leaves along the branches. Section 4 presents a discussion about recent state-of-the-art methods and open research lines. Finally, conclusions are listed in Section 5.

2 Background

In this section, we aim to provide an overview of previous studies focused on 3D tree reconstruction using real world data. Most of the reviewed studies follow a common pipeline, as shown in Fig. 1. In the literature, we can find a wide variety of methods that provide contributions in some of these stages from different approaches by considering measured data in real tree models. Input datasets may be represented in two dimensions (2D) or three dimensions (3D). Depending on the capturing procedure, the input data typically used for tree modeling are represented as follows: point clouds, which can be generated by both terrestrial or aerial Light Detection And Ranging (LiDAR) techniques [13], [14], [15] and photogrammetry [16], [17], [18], [19], a collection of multi-view images [20], a single RGB image [21] and depth maps [22], [23], [24], [25]. These data are usually simplified using a voxel or grid-based representation and indexed by spatial data structures such as octrees, quadtrees or kd-trees [26], [27], [28], [29]. The representation of input data can be adapted considering specific constraints for each approach. For instance, some methods based on guided modeling use biological rules to infer the 3D structure of a tree from 2D sketches [30], [31], [32]. For this purpose, key patterns are extracted from the images and different solvers are applied to detect the contours and shape of the target trees. More recently, other approaches rely on the use of 3D data such as point clouds and voxel representations to infer the modeling of tree skeletons. [33]. Moreover, novel experimental methods are proposed for sketch-based modeling of 3D trees in virtual reality [34], semantic segmentation of captured data to classify the trunk, branches and leaves by applying machine learning [35], [36], [37] and generating synthetic datasets [38].

After the preprocessing stage of input data, the resulting geometry is used to accurately model the 3D tree skeleton. Guided tree modeling from real-world data is the focus of this review that aims to collect and compare existing methods proposed to date. The body of previous work has been classified into three main categories:

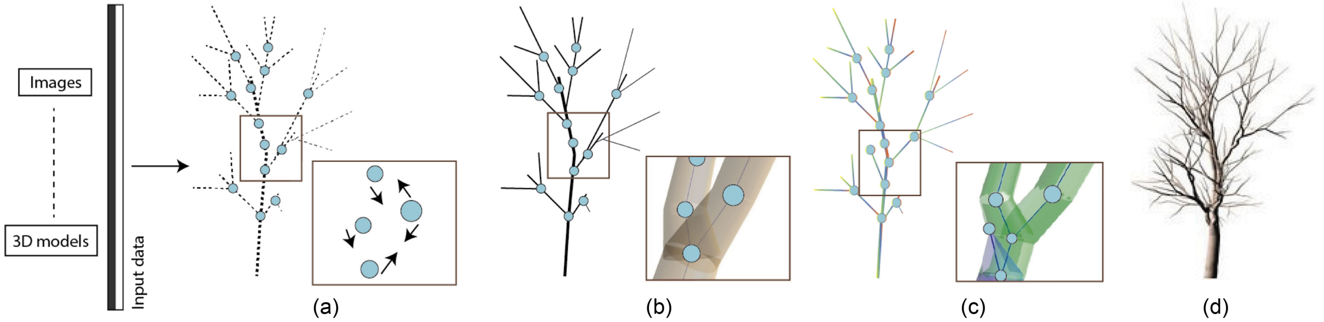


Fig. 2. Pipeline for the procedural reconstruction of the tree skeleton. (a) Point connection, (b) preliminary tree modeling, (c) refinement of the tree structure, (d) geometric rendering.

- *Procedural reconstruction*: a set of techniques is surveyed which are based on procedural rules, considering input data as a guide for the tree modeling.
- *Geometry-based extraction*: this set of methods aims to extract the topological skeleton, which resembles the appearance of the branching structure.
- *Image-based modeling*: These approaches focus on the generation of two-dimensional branches from image features for the estimation of the 3D tree skeleton.

Once the 3D tree skeleton is generated, the next step is the post-processing phase which integrates the following main techniques: (1) cleaning, (2) branch smoothing and (3) removal of redundant branches and (4) the change of straight segments into curves. A wide variety of methods focused on improving the final appearance of the tree to achieve a higher level of realism are also surveyed.

2.1 Procedural Reconstruction

Procedural modeling techniques have been widely used for the 3D modeling of trees. In recent decades, some efforts have been devoted to the procedural reconstruction of real trees. [39]. The methods proposed in this field of research can be classified into (a) guided modeling from real-world data (LiDAR scans, images, 3D data, etc.) [40], [41], [42] (b) interactive modeling using available frameworks such as Xfrog [43], [44], PlantGL [45] and Speed-Tree [46], and sketch-based approaches [30], [31], [32] and (c) fully autonomous procedural models [47], [48].

In this section, we aim to review procedural methods that take as input real-world data for tree modeling. This set of algorithms enables generating tree structures taking into account not only the geometrical and spatial constraints of the scanned tree structure, but also the surrounding entities of the environment that significantly determine the shape of the 3D model [41], [49]. In general, these methods follow the pipeline presented in Fig. 2. As input data, a sparse reconstruction of real-world trees is used. Then, 3D points are connected considering their properties as well as those of their neighbors. As a result, a preliminary branch-structure graph (BSG) is generated for each tree. After this process is completed, several optimization methods are usually applied to reconstruct the major skeletal branches of the scanned trees. Finally, finer structures such as leaves are synthesized from the BSGs to complete the reconstruction pipeline.

Procedural techniques for generating trees exploit the self-similarity property that makes their appearance fractal. Self-similarity happens when the approximate structure of the whole object is repeated at different scales, i.e., the structure of the trunk and main branches is like other branching ramifications of the same tree. This implies that they share a property called ”database amplification” [50]. It means that with a small set of rules, complex structures can be derived. When stochastic processes are introduced, the amplification factor enables the algorithm to produce many different models. Regarding the tree reconstruction task, methods based on this approach can benefit from the procedural generation of parts of the tree that are occluded, or the ones that have little information.

In this survey, two main sets of procedural methods are reviewed: (1) rule-based modeling and (2) particle-flows modeling. In rule-based systems, the structure of the tree is built from the root, using replacement rules that expand the tree geometry. This is a top-down approach where the most general parts are reconstructed first (the trunk and main branches), whereas in particle-based reconstruction the structure of the tree branching is generated from the trails of particles that fill the volume of the tree. The flow of the particles is simulated using several rules that define the particle motion. These methods can be considered a bottom-up counterpart to rule-based reconstruction: starting with the modeling of thinner branches and twigs and ending with the main ones and the trunk. Both approaches are described, and the main scientific contributions are presented in the following sections.

2.1.1 Rule-Based Modeling

In this section, rule-based methods for tree reconstruction are described. All of them have in common the presence of a rule system, which determines how to generate the branching structures of a tree. In order to achieve a real distribution of tree branches, the rules are set considering input data. This approach is known as inverse procedural modeling [51]. It is applied for stochastically generated trees from which polygonal 3D models are obtained and the parameters of a procedural model are estimated [41]. This is an extension to L-systems, which were introduced by Lindenmayer to model filamentous organisms [47].

L-systems are a form of rewriting system, containing a set of rules that describe replacements of symbols. Unlike

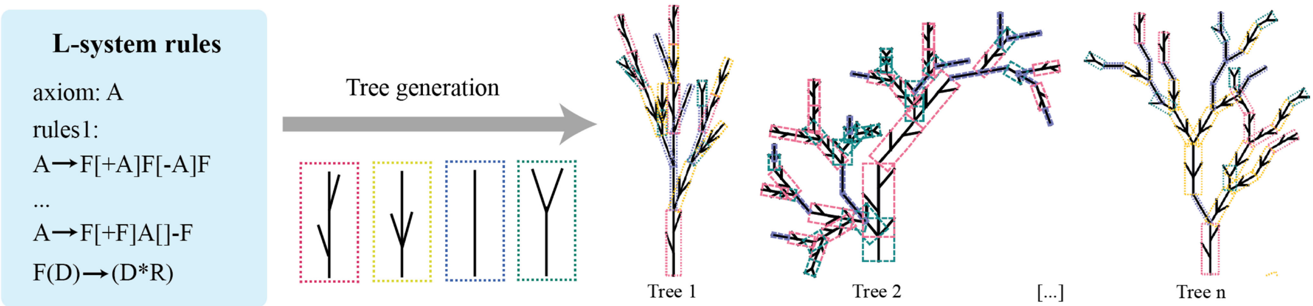


Fig. 3. Overview of rule-based modeling of the tree skeleton through the combination of some L-system rules.

formal language theory grammars, in L-systems, all production rules are applied simultaneously, so they become parallel rewriting grammars, and there is no distinction between terminal and non-terminal symbols [50]. L-systems work as follows: on each iteration, pieces of a string are replaced using derivation rules (Fig. 3). In the first iteration, this string is called axiom. The string represents a specific shape of a tree and this can be interpreted to retrieve the branching structure. Later, parametric L-systems introduced numerical parameters on the grammar symbols [52]. The values of these parameters are also replaced in each iteration, according to the rules. The book of Prusinkiewicz and Lindenmayer [53] is a compilation of all these techniques applied to plant modeling. Conceptually, the tree reconstruction used by these methods is closer to the variation of parametric L-systems proposed by Měch and Prusinkiewicz [54], called "Open L-Systems." Open L-systems are an extension to environmentally sensitive L-systems [55], which enable bidirectional interaction between the plant and the environment. When a symbol replacement is needed by a rule, there may be an exchange of information where the plant model updates the environment (i.e., changing the available space due to the growth of the tree) or the surrounding environment limits the development of plants (i.e., there is a human-made object that prevents the tree growth in some directions). This can be applied to reconstruction, in which the result of the L-system must be similar to the target tree.

Taking open L-systems as a starting point, Sakaguchi *et al.* [56] is the first to reconstruct trees from captured data. In that work, a voxel grid is generated from a set of images of a real-world tree. Then, a segmentation of the image background is performed and a voxel grid is built, which is filled by using simple branching rules. Shlyakhter *et al.* [57] used a similar approach. The main difference is the use of a simplified mesh representing the visual hull of the tree as input, from which the axiom to the L-System is obtained. For the extraction of this visual hull, they consider the silhouettes from segmented images of the tree. The generated mesh represents the boundary of the visual hull. A medial-axis extraction algorithm derived from the Voronoi tessellation is used to generate the main branches and trunk. The resulting skeleton is considered the axiom to the open L-system that synthesizes the rest of the branches. After each iteration, branches that lie outside the bounding mesh are pruned. This method works properly for the characterization of trees with dense foliage and a well-defined crown shape. However, learning the parameters of a procedural model is computationally expensive. To overcome this, other methods are focused on geometric learning and topological features of tree species by fitting to the collection of a probability model [58], [59].

The above-mentioned research enables the development of further methods that are based on rule-based generation to predict additional branches and check if they match the input data. Branches are commonly modeled as 3D cylinders with different sections. To check whether the branches are valid, input 3D point cloud and image features are used [60], [61]. Even though input data involve the development of guided or inverse modeling methods [51], some parts of the tree cannot be directly captured from the real world due to occlusion and insufficient sensor resolution. Consequently, other methods focused on modeling the BSG, considering key features of the tree environment. Runions *et al.* [62] proposed the Space Colonization algorithm, based on L-Systems. This is introduced as a guided procedural modeling technique, in which the user defines the shape of the tree to be generated. The user aims to place a set of markers around the space, towards which branches grow. In this study, the environment plays a key role in modeling tree structures. The search for markers in the surroundings of the node is usually limited by thresholds considering several angles and maximum distances between branches and nodes. After unfolding the branch from a node, the predefined surrounding markers are removed. The process ends when there are no markers left. The space colonization approach was later improved by Pałubicki *et al.* [48]. In this work, light propagation is also considered a key factor to determine the growth of branches.

Preuksakarn *et al.* [64] used the space colonization algorithm to extract the architecture of a tree, following a novel approach in which the skeletal structures of the plant are iteratively created according to the local density of input point clouds. A method locally adaptive to the levels of precision of the data is developed to combine a contraction phase and a local point tracking algorithm. Moreover, a great amount of research was carried out for 3D tree reconstruction by using scanning technology. Later, Xie *et al.* [65] proposed a method for tree modeling, in which tree slices were intuitively positioned in space, serving as efficient proxies that guided the creation of the complete tree. Allometry rules were considered to ensure reasonable relations between adjacent branches. Pirk *et al.* [66] converted this representation into a dynamic data structure that allows trees to react to their environment [67]. This research line was focused on the dynamic and realistic behavior of plant models and has recently aroused interest in tree growth [68],

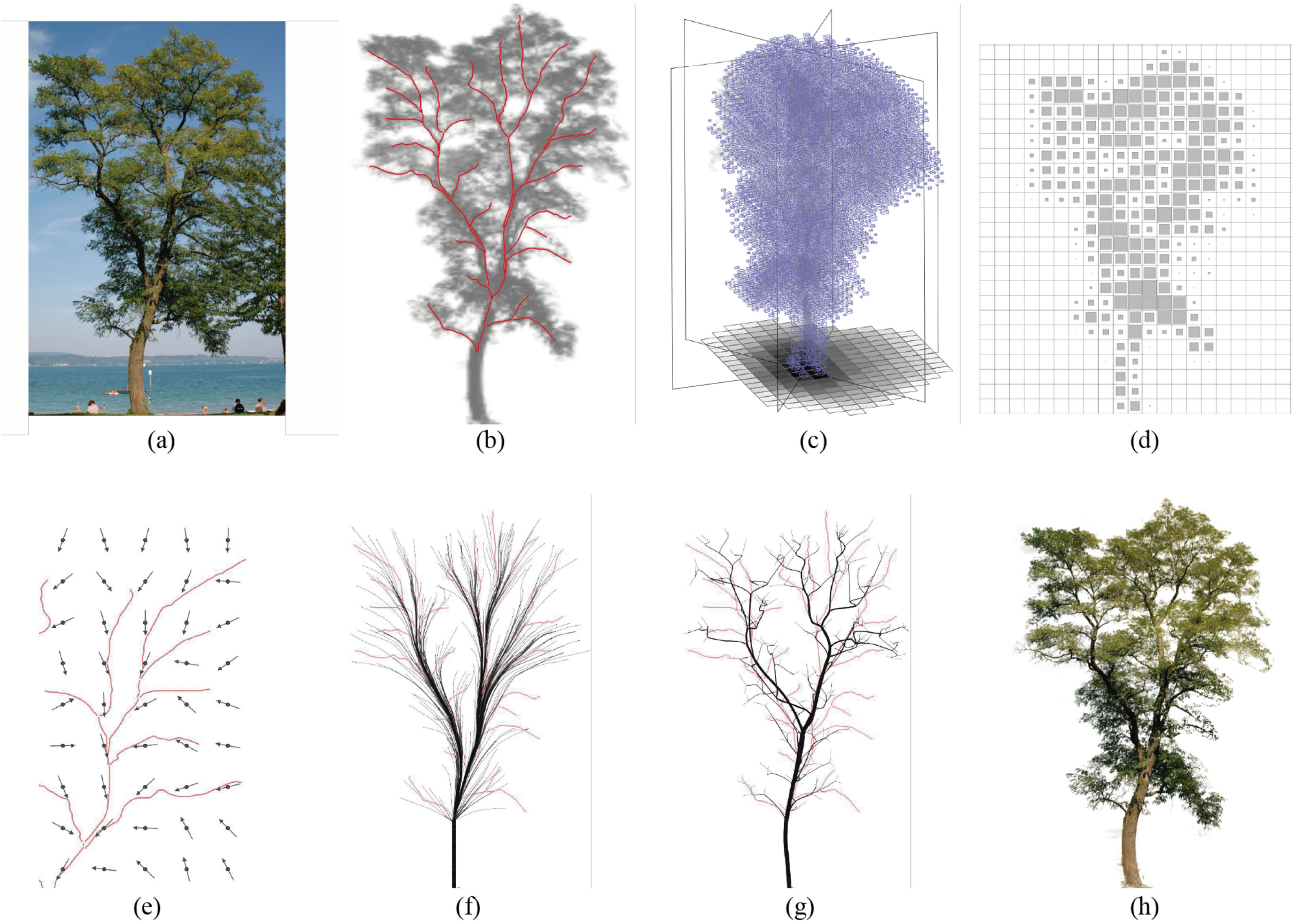


Fig. 4. Tree modeling using particle flows. (a) An input image, (b) tree density estimation with its corresponding attractor graph, (c) voxel grid by back-projection, (d) density values for one image plane, (e) estimation of direction fields for a graph, (f) result for linear blending without particle attraction, (g) result for linear blending including particle attraction, and (h) one example of resulting tree models [63].

the dynamic adaptation to support structures, as can be observed for climbing plants [69], [70] and modeling of realistic materials [71].

Recently, some contributions have been presented based on the use of input data and the space colonization algorithm by Guo *et al.* [42]. These advances are focused on the improvement of resulting tree structures by increasing the geometric similarity to target real-world trees. Beginning with several photographs from different viewpoints, the Structure from Motion (SfM) method is applied to generate dense point clouds [72]. For each 3D point, the normal and a score are calculated to estimate a similarity coefficient between the target point and its neighbors. These techniques work properly for leafless trees, but a high occlusion prevents the reconstruction of the branches. According to the resulting data by applying one or the other acquisition technique, these models are weighted. A greater weight is assigned to those surfaces of branches that are smoother and smaller weight to noisier parts near the crown. Then, the space colonization method is used, considering the computed weights: points with a greater weight attract the branch growth towards them since these correspond to well-defined branches. The maximum growth distance is reduced for each iteration if shorter branches are encountered at higher branch levels. Some parameters are needed, such as the branching angles, which depend on the target tree species and are not computed automatically.

2.1.2 Particle-Flows Modeling

This approach aims to place particles in three-dimensional space and set some physical rules for updating their position and speed. The resulting branching structures are formed by the trail of particles, connecting visible branches and synthesizing occluded ones. Particle systems have been widely used for modeling and rendering plants. Initially, Reeves *et al.* [73] introduced a particle system for modeling grass with a high level of realism at that time. They were later enhanced and used to model branches and leaves of multiple tree species [74]. Later, Rodkaew *et al.* [75] introduced new rules and physics focused on particle systems to simulate tree branches, roots and leaf venation patterns with the trails of the particles.

Fig. 4 presents a general overview of the main steps for tree modeling using particle flows by following the Neubert *et al.* [63] approach. Input images are usually used for modeling the tree structure. The first step is to separate the tree from the background. For this purpose, several methods aim to fill the uncertain regions with either opaque (foreground), transparent (background) or partly transparent pixels based on an interpolation, which considers the image gradients [76]. In a second step (Fig. 4b), the input image is used to sketch an estimated underlying tree skeleton. Then, the 3D plant volume is estimated from the input image by calculating density values for the voxels (Figs. 4c and 4d). A discretized version of the volume rendering equation enables the computation of the desired solution. According to particle tracing, density values are used to create initial particle positions for the main tree skeleton. The particles are randomly located in voxels considering their density. The particles are mainly directed towards the tree base and their respective nearest neighbors. The direction vectors are computed for a three-dimensional grid and as a result of applying a linear blending function, a 3D graph of the main branching is obtained (Fig. 4f). Then, Fig. 4g shows the result after fitting branching angles and applying the particle attraction to join particles and form the tree skeleton attraction. Finally, the 3D geometry of the tree is created by assigning for each branch a thickness value and introducing the tree foliage (Fig. 4h).

One of the main contributions in the particle-based tree reconstruction domain is the method proposed by Neubert *et al.* [63], which extends the volumetric reconstruction of Reche *et al.* [5] with particle flow to recover the skeleton of the tree. Following the approach of using billboards to render the volumetric representation of the tree [5], Neubert proposed the creation of a 3D grid in order to estimate the density value for each cell. The disadvantage of this method is the difficulty of finding the appropriate parameter settings for particle simulation. Also, it is not recommended to be applied with those trees which do not have the same structure on different branching levels. They must have a high self-similarity, as the differences in structure between levels are more difficult to encode in the particle updating rules.

Long and Jones [77] avoid the need to position the images at the start, placing markers at the end of the branches and using Motion-Capture to estimate the pose. They substitute the voxel grid with a set of bounding boxes stacked on the vertical axis, which contain the markers. Particles are generated inside these volumes using markers and random points. Zhang *et al.* [78] added some extensions to the method of Neubert, mentioned before [63]. Instead of using images, this method is based on a multi-layer extension of particle flows and uses input point clouds which are obtained from scanning. In the beginning, they apply a cylinder marching process to search for cylinders in the point cloud, which will be connected between them later by particle flows. Each layer represents a level of detail or branch order: first for the trunk and main branches, following the secondary, tertiary branches and twigs. This multi-layer particle flow also alleviates the problem by having the same structure at different levels to compute the branching structure. More recently, Isokane *et al.* [79] used the particle-flows reconstruction based on machine learning to build a grid from input images, so that each pixel stores the probability of being part of a branch of the tree structure. This method was based on statistical inferences of branches using a Bayesian extension of image-to-image translation applied for each multi-view image.

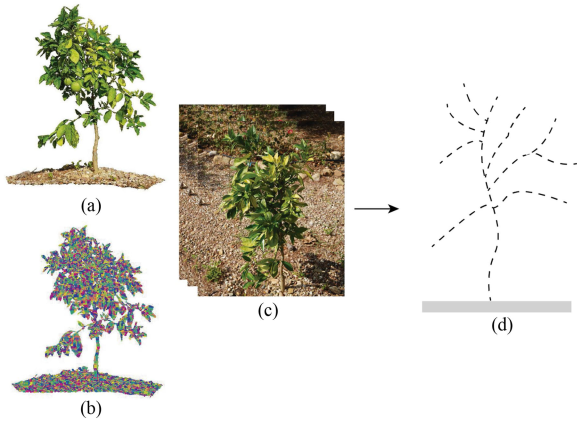


Fig. 5. The generation of 3D skeleton of a tree by processing input. (a) The 3D point cloud, (b) voxelization of a 3D model, (c) multi-view images and (d) the main branching structure.

2.2 Geometry-Based Extraction

In this section, previous work focused on processing input data (point clouds, voxels, images) for the extraction of the Medial Axis (MA) and the topological/morphological skeleton of trees is presented (Fig. 5). The concept of MA was introduced by Blum *et al.* [80] as a descriptor of the shape of any 2D object. For a planar surface, MA defines the loci of circles of maximal diameter that maintain contact with at least two points on the surface border. The Medial Axis Transform (MAT) is the medial axis together with the associated radius function for each maximal ball around any given point on the MA. Later, Sherbrooke *et al.* [81] proposed an algorithm for computing the Medial Axis Transform of a 3D polyhedral solid.

In the field of tree modeling, MAT is useful for tree reconstruction since the topology of the branching structure just can be obtained if the rest of the branches are also estimated. MAT allows us to represent the radius of tree branches at different points of the 3D tree skeleton. In the literature there are many methods focused on MAT computing for the generation of the tree skeleton [82] and the leaf surface [3], as shown in Fig. 5. Saha *et al.* [83] reviewed a wide variety of skeletonization algorithms for general purposes that were grouped into three main categories considering the underlying object representation: (1) algorithms based on Voronoi diagram or continuous geometric approaches of point clouds, polygonal representations of object boundaries, (2) algorithms based on the principle of the continuous evolution of object boundary curves and (3) algorithms based on the principle of digital morphological erosion or location of singularities on a digital distance transform (DT) field.

Generally, these methods aim to determine the topology of input data. However, captured 3D data is usually incomplete, which means that those approaches based on the medial-axis estimation have important limitations due to their high dependence on input data. Consequently, current methods based on medial axes extraction do not perform properly in dense-foliage trees, where the branch structure is occluded, as well as in small branches if captured data do not have enough resolution. Better results are obtained in defoliated trees since this approach is not based on generative techniques. Even though these methods perform worse by inferring missing or occluded parts in the tree body, the MA extraction is useful to estimate the trunk and first levels of main branches in a dense canopy. Then, MA may be used as a guide to model the tree geometry by a procedural technique [57]. Novel solutions have been presented to overcome this problem by considering the spatial distribution of surrounding main branches to the medial axes of a tree. In summary, this survey presents three categories, which are described below:

- *Thinning*: this category includes those methods based on both the morphological erosion and singularities of the distance transform of a voxel grid.
- *Clustering*: this category includes a set of methods that aim to extract groups of 3D points that represent the shape of main branching sections. Then, the centroids are joined to generate the branching structure.
- *Spanning tree refinement*: this category includes methods that are based on the construction of a spanning tree from a point cloud.

In the rest of this section, a brief survey of MA-extraction and tree skeletonization algorithms for each of the above three categories are presented.

2.2.1 Thinning

These methods are focused on computing the volume of the tree by generating a thin-based representation from the input data to preserve the topology of the tree shape. Then, this representation is used to create the tree skeleton. To ensure proper results, it is important to estimate the tree volume. On the one hand, both voxel/grid octrees are computed to determine the relationships between close branches, which are considered volumetric components. On the other hand, point clouds are processed to model the 3D surface and provide a volumetric representation of input data.

One way to apply the thinning process to the 3D tree structure is by contracting or eroding the surface of branches until the thinnest possible representation is achieved. For the volumetric representation of the tree, this process is applied to remove those parts of the tree volume, which do not belong to the medial axis. For bi-dimensional representations, the surface position is updated by contracting the trunk and branches. Another way is to compute the distance to the boundary for each element of the tree model. Finding local maxima in the distance transform yields the medial axis, as well as lines equidistant to the boundary.

Gorte and Pfeifer *et al.* [13] proposed a novel method for the MA extraction considering a voxel representation of scanned trees. First, they filled the hollow branches by combining dilation and erosion operations, as scanned data just represent some 3D points on the surface. Then, they applied the erosion operation to thin tree branches as much as possible. Finally, the topology was obtained by connecting voxels with their neighbors. As a limitation, this method can create only branching structures from scanned data of trees without leaves.

Later, Bucksch and Lindenbergh *et al.* [84] used structures such as the octree for the generation of the tree skeleton. This method was parametrized and some rules were defined to avoid infinite loops. Su *et al.* [85] applied an iterative process considering 3D point clouds as input data. A geometric simplification based on Laplacian smoothing was carried out by the estimation of the normal vector for each 3D point and obtaining an approximation to the surface of the branching structure. Then, some points of the main branches and the base of the trunk are fixed as anchors to ensure the real shape of the scanned tree. After obtaining the thinnest possible representation of input data, the point cloud is sub-sampled in order to reduce the skeleton complexity. This sub-sampling is adaptive (those parts with a higher detail contain more points). Finally, those branches which are near enough are merged. More recently, Bartolozzi and Kuruc *et al.* [86] applied a thinning process on an input 3D mesh and carried out the shortest path search in the graph from the root to the tips of the branches (defined by an angle close to 0 degrees) to avoid cycles. To avoid false parallel branches, they use an adaptive radius that shrinks in size as it gets from the trunk to the branches. Branches lying inside this radius have collapsed.

2.2.2 Clustering

This set of methods is based on pooling input data considering both spatial and semantic features. The resulting clusters are formed by several sections of tree branches, which are then connected in order to generate the tree skeleton. Generally, the first step is to estimate an approximate topology by calculating distances between nearby elements that probably belong to the same branch. Voxel-based grids and octrees allow us to determine implicit relationships between data since these are sorted spatially. On point clouds, the computation of such neighborhoods is achieved by building a geodesic graph which is generated by connecting spatially close points. Thus, branches are segmented into multiple sections which are then joined considering some spatial constraints. Xu *et al.* [87] presented a novel approach focused on extracting the skeleton of a tree by following the above-mentioned workflow. This is an adaptation of the skeletonization algorithm of point clouds presented by Verroust and Lazarus *et al.* [88]. The adaptation of this approach has been repeated by other authors later, and the main steps are briefly described below (Fig. 6). First, a raw point is ordered by an octree. The centroid for each octree cell is calculated. Second, a neighbor graph is generated considering the estimated centroids. Then, a root node is selected as the base of the trunk of the tree. Thus, a distance from each 3D point to the root node is calculated. As a result, a geodesic graph is obtained. Third, the 3D points are clustered by the distance considering a fixed number of intervals or a fixed interval length. Then, each cluster is divided into its connected parts. Thus, parts of the tree that belong to different branches are separated even though they have almost the same distance to the root node. Finally, the centroids for each cluster are connected for the generation of the tree skeleton. In this step, it is common to apply a refinement process to obtain the final skeleton. After presenting, the overview of the skeletonization method based on clustering, some significant considerations for each step are presented.

Neighborhood functions are usually based on the k-nearest neighbors (KNN) considering a search radius [20], [87], [89]. This distance may not allow us to generate a fully

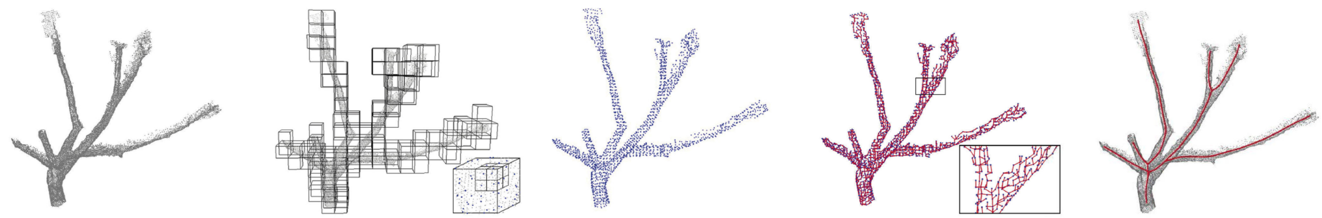


Fig. 6. Overview of the skeletonization method following a clustering approach. (a) Raw point cloud, (b) octree generated from the input point cloud, (c) the centroid of every octree cell, (d) The geodesic graph obtained from the neighbor graph and (e) the final skeleton after the refinement process [26].

connected graph, due to the occlusion. This long-standing problem has led to erroneous joins between branch sections, so several solutions were presented such as Delagrange *et al.* [89] that provided a manual method to mark occluded branches and Xu *et al.* [87] that solved it automatically by extending the tree skeleton to include those disconnected branches. This method defined a cone-based search at the end of each branch, considering a specific angle and a maximum distance to the branch orientation so that disconnected branches could be detected and connected to the rest of the tree structure.

The KNN function builds the neighborhood connecting a point with its k nearest neighbors [90], [91]. The value of k mainly depends on input data and this can be more difficult to tune than the radius, as it is less intuitive. If this is too low, the estimated graph can become disconnected and if this is too high, false branch connections can be generated. As an alternative, other neighborhood functions can be used which are supported by octrees or voxel grids. Fu *et al.* [26] developed an octree from the input point cloud and just considered cells of leaf nodes to compute the nearest neighbors by their adjacency.

In the stage of computing the geodesic graph and the distances to cluster branch segments, the most common distance function is the 3D euclidean distance between points, but other methods have been presented based on other functions. Tan *et al.* [20] assigned a weight for each section of the branch by using a custom function that mixed the 3D euclidean distance and the distance from the projection of points on captured images. Li *et al.* [35] used the number of nodes traversed as distance. Likewise, there are different clustering strategies, the most common being a binning of distances from every 3D point to the root of the tree. This binning operation aims at splitting a continuous variable into discrete categories forming ranges of said values. This operation can be defined by a fixed number of bins or by the range for each bin [91]. These ranges can have an equal size [20], [87] or their sizes decrease according to the distance from the root node [35]. This last one enables a branch to become shorter as its level on the tree gets further from the root. Other authors presented different clustering methods: Yan *et al.* [90] used a similar clustering approach to that proposed by Cohen-Steiner *et al.* [92], adapting it by a shape approximation of the estimated surfaces on the point cloud. Gao *et al.* [93] extracted geometric/spatial features from the point cloud, which were considered for the clustering process. Once the 3D points are clustered by distance, the branching sections that are in the same distance range are split into their connected components to generate different branches. This process poses two main challenges: some branching sections that are supposed to be connected are not connected, and vice versa. As a solution to this problem, instead of using the connectivity of the geodetic graph, Gong *et al.* [94] searched for contour lines of branches and connected them by a similar approach to the above-mentioned splitting and connecting process. Fu *et al.* [26] exploited prior cylindrical constraints of tree branches in order to center again the nodes of the skeleton and split some clusters that contain branch bifurcations into two separate clusters, despite they were connected by the graph. As the final step, all branching sections are connected to form the topology of the tree skeleton. For the last levels of branches, clustering-based methods present lower reliability, so other approaches are applied such as replicating some parts of the tree skeleton and using the auto-similarity property [20], [87]. Other methods proposed a point cloud segmentation in order to identify branches or leaves in the tree model. Thus, only those points of the branching structure are considered for the skeletonization method [35]. On the other hand, many existing methods do not segment the input point cloud. As a result, they do not achieve an optimal performance even when dealing with foliage trees. On the other hand, a more recent technique proposed by Liu *et al.* [95] used deep learning on point clouds [96] for the tree decomposition in tree branch segments by finding features on the point cloud. Instead of using the distance binning approach and graph, they made clusters of branches that were represented as generalized cylinders in two stages: first detecting small clusters and later merging them into larger ones that represent branch segments.

2.2.3 Spanning Tree Refinement

The afore-mentioned literature review shows that those techniques focused on the tree skeleton extraction have limitations if the input data are highly incomplete. To overcome this challenging problem, some methods aim to compute the minimum spanning tree (DMst) based on a global optimization method. This approach aims at constructing a distribution tree from input point clouds, whose edges are weighted by an estimated distance between the nodes of the tree. Thus, a weighted spanning tree is generated, which partially represents the 3D shape of a real-world tree. As the main novelty of this approach, the DMst is adaptive to variable point density to model tree skeletons from scanned point clouds. In general, the computation of the spanning tree can be approached (1) by calculating the geodesic graph that results from computing Dijkstra's algorithm from the

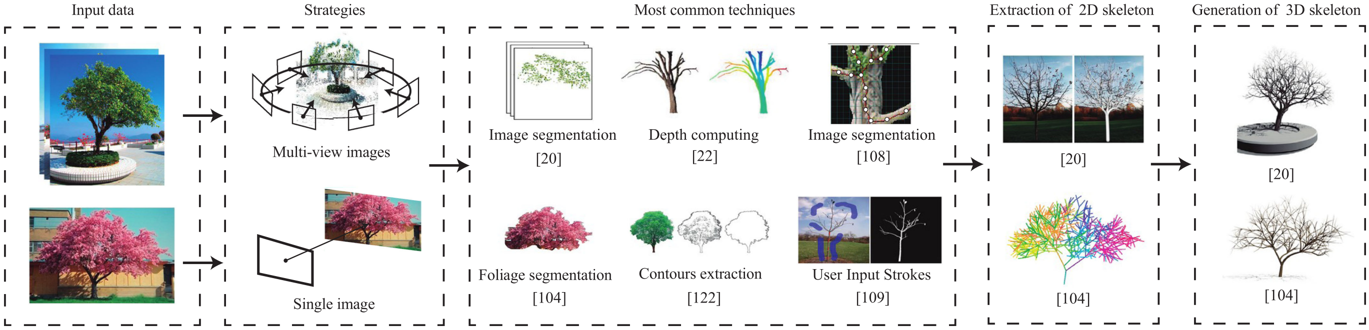


Fig. 7. Overview of image-based methods for tree modeling. Two strategies considering one or multiple images as input data are presented. Then, most common techniques and the extraction of 2D skeletons from images are shown. Finally, the resulting 3D skeleton are for each strategy are depicted.

root node [97] or (2) by calculating the DMst, which is a minimum weight spanning tree, where the sum of the weights of all the edges must be minimal [98], [99]. The resulting DMst is noisy and often contains erroneous branches. To remove them, a weight-based calculation proportional to the length from the root node to the position of the revised 3D point is performed. Therefore, false branches have lower weights than the trunk and main branches and these may be automatically removed. In addition, the scanned 3D points only represent the surface of the branch structure. Accordingly, they are created on the boundary and not on the medial axis. This issue is also considered by the simplification operations, mentioned above.

There are two main approaches for the spanning tree refinement: a global optimization [97], [99] and a local optimization [98]. On the one hand, Livny *et al.* [97] started by computing the tree of minimum distances to the root and determined a smoothed orientation field for the update of vertex positions. This process needs multiple iterations. Wang *et al.* [99] also use a global approach, based on the interaction between points with repelling forces. On the other hand, Du *et al.* [98] simplified the MST by removing those points with the lowest weight that are connected to others with greater weight, since these generally correspond to noise branches. Then, if the distance between two points is lower than a predefined threshold, both are merged (local optimization).

2.3 Image-Based Modeling

This set of methods can be applied through two main approaches. The first one is based on the use of multi-view images to generate 3D point clouds by applying the SfM algorithm [72]. The second relies on a voxel-based strategy to generate 3D data because fewer images are available [100], [101]. If many overlapped images are available as input data, the augmentation of the 3D data may be performed by using the projected distance in pixels for a 3D graph, and textures to characterize the leaves and trunk [20]). The voxel-based algorithms are mainly focused on the extraction of 2D skeletons from the images.

Both approaches use images as input data for the extraction of the tree skeleton. Initially, the 2D skeleton is obtained from single or multi-view images. Then, this 2D representation is projected to a 3D space by different strategies in order to extract the 3D tree skeleton. In general, input images are segmented by applying semi-automatic methods. The skeleton extraction process is similar to the ones used in 3D, especially those that manage voxel-based models. In this way, the input voxels are replaced by pixels of the image. Those methods that only extract the 2D skeleton of the image [102] are out of the scope of this survey, as we are focused on the extraction of the 3D tree skeleton. Most of them resemble those exposed in Section 2.2, but they operate with 2D data.

Fig. 7 shows an overview of image-based tree modeling. The extraction of the 3D skeleton is validated by mapping it on input images and following a qualitative or quantitative assessment. Cheng *et al.* [22] used a range of images as input and used depth information to estimate the angles between branches in the 3D space. Depth data help to achieve higher accuracy for the estimation of branch placement. According to the use of a single photo or multi-view image for the skeleton extraction, two sets of methods are presented.

On the one hand, multiple methods based on single image tree modeling have been proposed. Zeng *et al.* [103] presented an approach for the 3D reconstruction of tree models using only a single image of trees without leaves, which is segmented from the background. Later, Tan *et al.* [104] segmented the input image to extract the foliage region and the branching structure on the image. This method does not intend to recover 3D structure directly from the image. In contrast, they used the image as a guide for non-parametric tree growing, i.e., the growth should lead to a result close to the image. Different from sketching, they only drew a few strokes. The image statistics underlined by the strokes enabled the recovery of more tree structures. More recently, Guenard *et al.* [105], [106] allowed to generate a possible skeleton from a foliage segmentation. A 3D generative model was carried out by following a parametric model that considers botanical knowledge. More recent techniques based on neural networks have been proposed for the reconstruction of 3D trees from a single image. Liu *et al.* [107] used a similar approach to previously mentioned, but instead of maximizing the angle of projection, they used Generative Adversarial Networks (GAN) to learn the reversal of the projection from 3D to 2D. Based on the input images for each tree species, the 3D radial boundary volume is calculated. This data structure encodes the tree shape as a set of cylindrical layers of sectors. To reconstruct the 3D branching structure, they used a rule-based system.

On the other hand, the second strategy aims to extract the 3D skeleton by fusing the 2D branched structure resulting from all the multi-view images. The resulting skeleton is more precise since the tree shape is observed from different viewpoints. Teng and Chen *et al.* [108] proposed an image-based method for generating a complete 3D model of a tree. Some processes such as image segmentation, trunk point connection, and parameter selection require user interaction to complete them. Initially, the 2D skeleton was estimated and after the 2D trunk points were connected, recovering the 3D trunk skeleton is equivalent to estimating the 3D interpretations of the 2D trunk points, namely, the 3D trunk points. Because the camera was calibrated, the 3D trunk points were estimated by triangulation. Specifically, each 2D trunk point in an image plane represented one line in the 3D space. Subsequently, Lopez *et al.* [109] obtained the branching structure using the distance transformation on the image space. First, they estimated the 2D skeleton graph from its matte image and then found a 2D skeleton tree from the graph by imposing tree topology. For each edge in the 3D skeleton tree, they further applied a volumetric reconstruction to recover its corresponding curved branch. Finally, they used pieces of cylinders to approximate each branch from the volumetric result.

3 Extensions and Post-Processing

In this section we discuss some techniques used to extend the tree skeleton extraction process. These methods are intended to enhance the tree skeleton by adding additional features to achieve a realistic appearance of the tree models. In addition, some solutions are provided to address challenging problems in the reconstruction stage.

3.1 Dealing With Occlusion

The computation of occlusion in data captured from tree models is a challenging problem. Many of methods usually take as input dense point clouds, which are generated by LiDAR or photogrammetry. Nature is complex to be represented and usually a high occlusion level happens due to overlapped branching and dense foliage. Despite that, current acquisition systems such as UAVs which integrate high-resolution cameras [110], [111] and LiDAR [112], [113] and terrestrial laser scanners [114], [115], [116], [117] allow us to capture multiple images from multiple view angles, there are many branches which cannot be directly visible, specifically into the tree canopy. This is the reason why most methods only work properly on trees without leaves. If the foliage has a low density, only some sections of branches are not visible. In these cases, the tree structure is recoverable, but the reconstruction process may be challenging.

On the one hand, if the trees are characterized by dense foliage and only several branches of the real model can be extracted, methods based on procedural modeling provide more adequate results than geometry-based extraction techniques. They can generate plausible tree structures if rules are appropriate for the specimen by inferring on the first levels of branches through the extraction of the medial axis [57]. However, there is no way to evaluate the quality of the reconstruction, except visually. If the top of the tree is occluded because of taking images only from the ground, Guo *et al.* [42] computed the convex hull of the tree crown and filled empty parts with markers, so the L-system based growing rules could fill this space with branches.

On the other hand, if many branches are missing, the occlusion method proposed by Xu *et al.* [87] is more suitable for leafless trees. Wang *et al.* [99] used a global optimization as a way to update the point cloud and infer the correct joints between branches. Finally, a technique proposed by Isokane *et al.* [79] used deep learning in the image space to infer parts of the branching structure, which are occluded behind leaves. Occlusion is solved by a bi-dimensional approach, to the image space, before the 3D reconstruction of the tree skeleton.

3.2 Skeleton Refinement

Most tree modeling methods develop post-processing techniques on the tree skeleton after 3D reconstruction. The most common refinement operations are briefly described below.

Duplicated Branch Collapsing. This problem occurs especially in methods that take point clouds as input data. It appears when one branch is detected as two different parallel branches but it should be only one branch, so that, false parallel branches are created. As a solution, these parallel branches are merged as a single one considering the distance between each other and the radius of both branches. Thus, the overlapped volumes of branches are directly collapsed [42], [97].

Simplification. Some methods of the thinning and DMst categories generate a noisy skeleton with many short segments and incorrect branches, due to the nature of the applied algorithms. Then, these segments are simplified without reducing the resolution of the 3D model. As an example, Du *et al.* [98] used Douglas-Peucker's method [118] to reduce the complexity of the generated skeleton. Another problem is the occurrence of false tips, where small branches are detected as noise. These can be removed by weighting the skeleton by distance and removing those with a low weight.

Smoothing. This operation aims to replace straight line segments by using curves. Most methods are based on the skeleton extraction as straight segments which do not fit properly to the real tree structure. The branches of real trees present more organic forms, which are better approximated by curves.

3.3 From the 3D Skeleton to Tree Geometry

In this section, we present some methods focused on the tree geometry generation from the topology of the branching structure. Boudon *et al.* [119] reviewed many methods to model the branching structure as geometry for rendering, such as the cone-sphere where nodes are considered spheres that are joined by truncated cylinders tangent to their surface (implicit surfaces or isosurfaces). A subdivision of surfaces is carried out in which a coarse surface is smoothed and refined and generalized cylinders are created by sweeping a variable radius circle along a curve.

Generalized cylinders are widely used by many methods for representing reconstructed tree skeletons [42], [94], [97]. In practice, they are used in a discrete form by sampling the skeleton at intervals to generate contours taking into account the radius. Then, they are connected to each other.

TABLE 1
Plot of a Selection of Input/Category Methods Reviewed in Previous Sections

		Point Cloud	Images	Mesh
Procedural Modeling	Rules	Sakaguchi [56], Binney et al. [60], Preuksakarn et al. [64], Gong et al. [94]	Huang et al. [61], Guo et al. [42]	Shlyakhter et al. [57]
	Particle flows	Zhang et al. [78]	Neubert et al. [63], Long et al. [77], Isokane et al. [79]	
Geometric Extraction	Thinning	Gorte and Pfeifer [13], Bucksch and Lindenbergh [84], Su et al. [85]		Bartolozzi and Kuruc [86]
	Clustering	Xu et al. [87], Yan et al. [90], Zhu et al. [91], Delagrange et al. [89], Huang et al. [124], Li et al. [35], He et al. [33], Gong et al. [94], Gao et al. [93], Su et al. [125], Fu et al. [26], Liu et al. [95]	Tan et al. [20]	
	Spanning Tree Refinement	Livny et al. [97], Du et al. [98], Wang et al. [99]		
Image-based	Single Image		Zeng et al. [103], Cheng et al. [22], Tan et al. [104], Guénard et al. [105], [106], Argudo et al. [126], Li et al. [127], Liu et al. [107]	
	2D Fusion		López et al. [109], Li et al. [128]	

There are two approaches for the estimation of the branch radius. The first one applies allometric rules and the other one extracts the radius of branches from the real tree. Using allometric rules allows us to assign for each node a proportional radius to the height of tree branches. The use of allometric rules is useful for modeling occluded branches whose radius is not available to be captured. However, these approaches are not accurate with real-world trees, as they only set the radius ratio for ideal conditions. The radius of the branches estimation from the data depends on the method used to extract the skeleton: methods based on clusters can fit cylinders to the clusters [26], [90] or use the standard deviation [20], so estimation of the radius at various points of the skeleton is the one computed for the cluster. In other approaches, the cylinder adjustment is performed at the skeleton extraction method [60], [78], thus, the radius of each branch is fitted to the corresponding cylinder. If the cylinder fitting is not applied in this step, it can be carried out later by sampling segments of the skeleton [98].

3.4 Leaves and Crown of Trees

First, there are methods that extract the geometric shape of the leaves from a 3D point cloud at a high level of detail. Quan *et al.* [120] obtained a dense point cloud from a collection of images and computed the k-nearest neighbors for each point using the euclidean distance. Then, a graph is generated and divided into multiple partitions, so that each one is considered a leaf. The partitioning criteria are determined by the weight of each edge by applying a function that considers the euclidean distance in 3D, the distance in the image projections and the color difference of projected pixels. This weight is intended to be proportional to the probability of two connected points belonging to the same leaf. Generally, a 3D model to represent a leaf is chosen as a prototype and textures are obtained from input images. Bradley *et al.* [121] tried to fit clones of a deformable leaf model into the point cloud using RANSAC and the Iterative Closest Point algorithm (ICP) [122], [123]. For each fitted leaf, their corresponding points are removed for subsequent iterations. In most cases, if the input point cloud is accurately segmented into leaves and branches, the reconstruction method provides more accurate results.

Second, other approaches are based on searching the leaf points to place the oriented leaf models there [20], [87]. If a segmentation of the tree is developed to achieve a more accurate 3D reconstruction of the branching structure, points corresponding to the leaves are commonly used to estimate the foliage density and the placement of leaves in the final model.

TABLE 2
A Comparison for Each Set of Methods According to the Sensitivity (1-4) of the Most Prominent Features That Influence the 3D Tree Modeling

Category	Subcategory	Criteria		
		Parameter tuning complexity	Foliage density supported	Tolerance to missing branch pieces
Procedural modeling	Rule-based	XXX	XXX	XX
	Particle-based	XXXX	XXXX	XXX
Geometric extraction	Thinning	X	X	X
	Clustering	XX	X	X
	Spanning tree	X	X	XX
Image-based	Single image	XX	XXX	X
	Image fusion	XX	XXX	XX

There is a third category of methods that aim to place leaves only following specific rules, without considering the foliage density or leaf position in the real tree [42], [97]. They do not intend to reconstruct the foliage, only to add leaves for a realistic rendering and optimal visual results. If target trees present dense foliage, it is not worth it to generate the skeleton structure. In these cases, there are significant contributions in which the crown shape is modeled, like the one proposed by Zhu *et al.* [91]. In that work, the alpha shapes are computed to model the tree canopy. In addition, Argudo *et al.* [21] modeled the tree crown from a single image, inferring its 3D shape from the 2D shape, and extracting the texture.

4 Discussion

Tree modeling has been a long-standing problem in computer graphics. The reconstruction of vegetation in real-world scenarios poses many challenges that previous research has approached. In this article, we reviewed the main existing approaches in the literature that focused on the reconstruction of the tree branching structure from input data. The aim was to provide the reader with a general overview of the topic, with all existing methods found in literature classified into categories. Moreover, we have collected other methods focused on the enrichment of the tree skeleton and improving the appearance of the reconstructed geometry. The reconstruction methods have been classified into several categories depending on the characteristics of the proposed algorithms for modeling the skeleton of the trees: (1) procedural reconstruction, (2) geometric extraction techniques and (3) image-based methods. For each category, the main contributions have been highlighted and compared to each other. A visual compendium of the presented methods is shown in Table 1.

Regarding the input data and pipelines, every approach has advantages and drawbacks. One of the main contributions of this survey is to help researchers to choose the most adequate methodology depending on the application. In Table 2 we summarize a comparison between different categories considering some crucial criteria that influence the results of tree modeling.

On the one hand, procedural-generative techniques are based on a tree growth pattern. In general, these can model challenging structures with missing pieces of branches. Specifically, the particle-based approaches usually give better results than rule-based methods since particles are tracked from the tips of the branches to the base of the trunk, so they are guaranteed to reach complete branching structures. In any case, rule-based approaches need to be tuned by increasing the search radius for the growth of a given branch or including some additional mechanisms to deal with those missing branch pieces. In other cases where trees have multiple trunks growing from the ground, rule-based methods should have their rules adapted for such a scenario. As advantages of the procedure-based techniques, they allow the reconstruction of trees with high foliage density due to their generative nature, filling the space with branches using tree growth (in the rule-based ones) or tracing particle trajectories (in the particle-based ones). Moreover, they can generate spurious branches due to outliers and noise in the input data. Another disadvantage is a large number of parameters required and the complexity of their setting, as they change depending on the tree species. Rule-based methods rely on many features that can be extracted directly from the tree shape such as the length of branches and the angles between them. Particle-based methods have the main drawback that the parameter setting is less intuitive. They describe the motion of the particles, so they do not translate well into measurable properties of the tree as in rule-based approaches.

On the other hand, the results derived from the geometry-based extraction have a high dependence on the input data and a low-level inference can be achieved. Therefore, if there are many missing parts along the tree body, these cannot be fully modeled through the base approach, and this has to be extended. Those methods are based on either thinning or clustering, the connectivity must be repaired using techniques like the ones described in Section 3.1. Spanning tree refinement algorithms tend to perform better at repairing connectivity since a spanning tree implies a single connected component in the graph. However, these approaches fail in most cases when the absence of geometry between branches is too high. [78]. Accordingly, they usually perform worse in those tree species which are characterized by dense foliage. In general, this set of methods requires fewer parameters than procedural-based ones, since the former needs the parameters of the tree model (maximum branching angles, branch segment length) for generating branches and fitting them to the data. The next set of methods is based on imaging techniques. The main advantage is the ease of creating a bi-dimensional tree skeleton using image features. However, this approach has some limitations in generating the 3D tree skeleton as input images contain only 2D features. The recent advances in the production of high-resolution cameras and versatile platforms allow us to capture both overlapped images and scanned data from multiple viewpoints. Therefore, a preliminary tree skeleton can be directly obtained after applying fast-forward data processing. In single-image based reconstruction techniques, the main advantage is the ease with which data is obtained, but these are not as accurate as the other categories.

The generation of the tree structure is a fundamental task with multiple applications in a multidisciplinary domain. To date, multiple approaches have been proposed that focus on modeling branching structures by using real-world data. After generating the tree skeleton, the next goal is to get a real appearance by adding leaves. Thus, the tree would seem real, and its geometry would represent both the branching of the branches and the foliage. More recent research opens the development of a promising method based on deep learning through the generation of real-world datasets [129], where trees are generated following a procedural approach. Moreover, López *et al.* [130] simulated scanned point clouds by applying aerial and terrestrial acquisition plans. More examples of datasets in recent studies are computer-generated, as huge data amounts are needed for training neural networks [95]. In this field, there are still several challenges to be overcome, including the selection of metrics for the quantitative evaluation of the results in comparison with reference tree models in the real world.

5 Conclusion

Tree modeling is a challenging task that has been addressed by applying a wide variety of methods. However, there are open problems to be addressed in the future. The visual assessment of the results may be enough, where the reconstruction of trees is intended for decorative purposes, such as virtual environments, videogames and simulators, etc. There are only a few methods that proposed quantitative validations based on similarity measures of virtual and real skeletons [69]. In this scope, synthetic trees can be used for the validation procedure, as they are the only way to have ground truth for a given list of tree species. Likewise, the tree geometry can be generated from a given skeleton using any of the surveyed techniques. A feasible future work is to quantitatively compare the main tree reconstruction methods to see how they behave in different real scenarios.

Another interesting advance is the use of deep learning in some steps of the tree skeletonization process. There have been some recent methods, such as inferring occluded parts of trees or the reconstruction of rule sets in 2D L-systems. This research has been mainly based on image processing due to the high availability of large sets of images, where many tree species are collected. However, the advances in sensors and data acquisition allow us to generate more and more dense point clouds which are useful for guided modeling of trees. The advantages of learning-based methods enable working with incomplete data. Accordingly, current algorithms might be able to process the occluded geometry by inferring the missing branches. In addition, the accuracy of existing algorithms continues to improve, making them more reliable in noisy conditions [26].

The rise of versatile capturing platforms such as drones and cameras allows us to obtain real-world data and cover complex natural areas from multiple viewpoints. The capturing of high-resolution images and dense 3D scanned data of natural environments is becoming easier. Consequently, a huge dataset will be acquired that requires novel methods based on reducing and cleaning useless data and massive processing with 3D data. In addition to the extraction of individual tree skeletons, such geometry is generally influenced by the environment, therefore, future research will focus on modeling and rendering dynamic ecosystems [9], [10], [131] and interactive applications [132]. Current advances are being developed in this direction to exploit real-world data based on neural networks [133]. Furthermore, there should be some advances in 3D plant modeling focused on pattern recognition in ultra-high resolution images, efficiency for the reconstruction process and other crucial optimizations based on parallelization by graphics processing units.

References

[1] S. Ulam, “On some mathematical problems connected with patterns of growth of figures,” in *Proc. Symposia Appl. Math.*, 1962, vol. 14, pp. 215–224.

[2] S. I. Sen and A. M. Day, “Modelling trees and their interaction with the environment: A survey,” *Comput. Graph.*, vol. 29, no. 5, pp. 805–817, Oct. 2005.

[3] L. Mundermann, P. MacMurchy, J. Pivovarov, and P. Prusinkiewicz, “Modeling lobed leaves,” in *Proc. Comput. Graph. Int.*, 2003, pp. 60–65.

[4] G. Bao, H. Li, X. Zhang, W. Che, and M. Jaeger, “Realistic real-time rendering for large-scale forest scenes,” in *Proc. IEEE Int. Symp. VR Innov.*, 2011, pp. 217–223.

[5] A. Reche-Martinez, I. Martin, and G. Drettakis, “Volumetric reconstruction and interactive rendering of trees from photographs,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 720–727, Aug. 2004.

[6] J. Stam, “Stochastic dynamics: Simulating the effects of turbulence on flexible structures,” *Comput. Graph. Forum*, vol. 16, no. s3, pp. C159–C164, 1997.

[7] S. Pirk, T. Niese, T. Hädrich, B. Benes, and O. Deussen, “Windy trees: Computing stress response for developmental tree models,” *ACM Trans. Graph.*, vol. 33, no. 6, pp. 204:1–204:11, Nov. 2014.

[8] S. Pirk, M. Jarząbek, T. Hädrich, D. L. Michels, and W. Palubicki, “Interactive wood combustion for botanical tree models,” *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–12, Nov. 2017.

[9] T. Hädrich, D. T. Banuti, W. Pałubicki, S. Pirk, and D. L. Michels, “Fire in paradise: Mesoscale simulation of wildfires,” *ACM Trans. Graph.*, vol. 40, no. 4, Jul. 2021, Art. no. 163.

[10] P. Ecormier-Nocca *et al.*, “Authoring consistent landscapes with flora and fauna,” *ACM Trans. Graph.*, vol. 40, 2021, Art. no. 105.

[11] M. Makowski, T. Hädrich, D. Michels, S. Pirk, W. Palubicki, and W. Skowronska, “Interactive simulation of 3-D ecoclimates,” in *Proc. EGU Gen. Assem. Conf. Abstr.*, 2021, pp. EGU2116–600.

[12] W. Pałubicki, M. Makowski, W. Gajda, T. Hädrich, D. L. Michels, and S. Pirk, “Ecoclimates: Climate-response modeling of vegetation,” *ACM Trans. Graph.*, vol. 41, no. 4, Jul. 2022.

[13] B. Gorte and N. Pfeifer, “Structuring laser-scanned trees using 3D mathematical morphology,” in *Proc. Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, 2004, pp. 929–933.

[14] D. Jaskierniak *et al.*, “Individual tree detection and crown delineation from unmanned aircraft system (UAS) LiDAR in structurally complex mixed species eucalypt forests,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 171, pp. 171–187, Jan. 2021.

[15] D. Mongus and B. Žalik, “An efficient approach to 3D single tree-crown delineation in LiDAR data,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 108, pp. 219–233, Oct. 2015.

[16] J. Lisein, M. Pierrot-Deseilligny, S. Bonnet, and P. Lejeune, “A photogrammetric workflow for the creation of a forest canopy height model from small unmanned aerial system imagery,” *Forests*, vol. 4, no. 4, pp. 922–944, 2013.

[17] D. Gatziolis, J. F. Lienard, A. Vogs, and N. S. Strigul, "3D tree dimensionality assessment using photogrammetry and small unmanned aerial vehicles," *PLoS One*, vol. 10, no. 9, 2015, Art. no. e0137765.
[18] P. Surovy, A. Yoshimoto, and D. Panagiotidis, "Accuracy of reconstruction of the tree stem surface using terrestrial close-range photogrammetry," *Remote Sens.*, vol. 8, no. 2, 2016, Art. no. 123.
[19] D. Lee *et al.*, "Analysing forests using dense point clouds," in *Proc. Int. Conf. Image Vis. Comput.*, 2018, pp. 1–6.
[20] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan, "Image-based tree modeling," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 87–es, Jul. 2007.
[21] O. Argudo, A. Chica, and C. Andujar, "Single-picture reconstruction and rendering of trees for plausible vegetation synthesis," *Comput. Graph.*, vol. 57, pp. 55–67, Jun. 2016.
[22] Z.-L. Cheng, X.-P. Zhang, and B.-Q. Chen, "Simple reconstruction of tree branches from a single range image," *J. Comput. Sci. Technol.*, vol. 22, no. 6, pp. 846–858, 2007.
[23] L. Lou, Y. Liu, J. Han, and J. H. Doonan, "Accurate multi-view stereo 3D reconstruction for cost-effective plant phenotyping," in *Proc. Int. Conf. Image Anal. Recognit.*, 2014, pp. 349–356.
[24] Y. Lu, Y. Wang, Z. Chen, A. Khan, C. Salvaggio, and G. Lu, "3D plant root system reconstruction based on fusion of deep structure-from-motion and IMU," *Multimedia Tools Appl.*, vol. 80, pp. 17315–17331, 2021.
[25] T. T. Santos, L. V. Koenigkan, J. G. A. Barbedo, and G. C. Rodrigues, "3D plant modeling: Localization, mapping and segmentation for plant phenotyping using a single hand-held camera," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 247–263.
[26] L. Fu, J. Liu, J. Zhou, M. Zhang, and Y. Lin, "Tree skeletonization for raw point cloud exploiting cylindrical shape prior," *IEEE Access*, vol. 8, pp. 27327–27341, 2020.
[27] T. Hinks, H. Carr, L. Truong-Hong, and D. F. Laefer, "Point cloud data conversion into solid models via point-based voxelization," *J. Surveying Eng.*, vol. 139, no. 2, pp. 72–83, 2013.
[28] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via laplacian based contraction," in *Proc. Shape Model. Int. Conf.*, 2010, pp. 187–197.
[29] C. Cao, M. Preda, and T. Zaharia, "3D point cloud compression: A survey," in *Proc. 24th Int. Conf. 3D Web Technol.*, 2019, pp. 1–9.
[30] M. Okabe, S. Owada, and T. Igarash, "Interactive design of botanical trees using freehand sketches and example-based editing," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 487–496, 2005.
[31] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang, "Sketch-based tree modeling using markov random field," in *Proc. ACM SIGGRAPH Asia*, 2008, pp. 1–9.
[32] S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz, "TreeSketch: Interactive procedural modeling of trees on a tablet," in *Proc. Eurographics Workshop Sketch-Based Interfaces*, K. Singh and L. B. Kara, Eds., 2012, pp. 107–120.
[33] G. He, J. Yang, and S. Behnke, "Research on geometric features and point cloud properties for tree skeleton extraction," *Pers. Ubiquitous Comput.*, vol. 22, no. 5/6, pp. 903–910, Oct. 2018.
[34] Q. Yuan and Y. Huai, "Immersive sketch-based tree modeling in virtual reality," *Comput. Graph.*, vol. 94, pp. 132–143, 2021.
[35] R. Li, G. Bu, and P. Wang, "An automatic tree skeleton extracting method based on point cloud of terrestrial laser scanner," *Int. J. Opt.*, vol. 2017, 2017, Art. no. 5408503.
[36] J. M. Jurado, M. I. Ramos, C. Enríquez, and F. R. Feito, "The impact of canopy reflectance on the 3D structure of individual trees in a mediterranean forest," *Remote Sens.*, vol. 12, no. 9, Jan. 2020, Art. no. 1430.
[37] J. M. Jurado, J. L. Cárdenas, C. J. Ogayar, L. Ortega, and F. R. Feito, "Semantic segmentation of natural materials on a point cloud using spatial and multispectral features," *Sensors*, vol. 20, no. 8, 2020, Art. no. 2244.
[38] A. Chaudhury, F. Boudon, and C. Godin, "3D plant phenotyping: All you need is labelled point cloud data," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 244–260.
[39] M. N. Favorskaya and L. C. Jain, "Realistic tree modelling," in *Handbook on Advances in Remote Sensing and Geographic Information Systems*. Berlin, Germany: Springer, 2017, pp. 181–202.
[40] B. Beneš, O. Štava, R. Měch, and G. Miller, "Guided procedural modeling," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 325–334, 2011.
[41] O. Stava *et al.*, "Inverse procedural modelling of trees," *Comput. Graph. Forum*, vol. 33, no. 6, pp. 118–131, Sep. 2014.
[42] J. Guo, S. Xu, D.-M. Yan, Z. Cheng, M. Jaeger, and X. Zhang, "Realistic procedural plant modeling from multiple view images," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 2, pp. 1372–1384, Feb. 2020.
[43] B. Lintermann and O. Deussen, "Interactive modeling of plants," *IEEE Comput. Graph. Appl.*, vol. 19, no. 1, pp. 56–65, Jan./Feb. 1999.
[44] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz, "Realistic modeling and rendering of plant ecosystems," in *Proc. 25th Annu. Conf. Comput. Graph. Interactive Techn.*, 1998, pp. 275–286.
[45] C. Pradal, F. Boudon, C. Nouguier, J. Chopard, and C. Godin, "PlantGL: A python-based geometric library for 3D plant modelling at different scales," *Graphical Models*, vol. 71, no. 1, pp. 1–21, Jan. 2009.
[46] Q. Xiong and X.-Y. Huang, "Speed tree-based forest simulation system," in *Proc. Int. Conf. Elect. Control Eng.*, 2010, pp. 3033–3036.
[47] A. Lindenmayer, "Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs," *J. Theor. Biol.*, vol. 18, no. 3, pp. 300–315, Mar. 1968.
[48] W. Palubicki *et al.*, "Self-organizing tree models for image synthesis," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 1.
[49] M. Makowski, T. Hädrich, J. Scheffczyk, D. L. Michels, S. Pirk, and W. Pałubicki, "Synthetic silviculture: Multi-scale modeling of plant ecosystems," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019, Art. no. 131.
[50] A. R. Smith, "Plants, fractals, and formal languages," in *Proc. 11th Annu. Conf. Comput. Graph. Interactive Techn.*, 1984, pp. 1–10.
[51] J. Guo *et al.*, "Inverse procedural modeling of branching structures by inferring L-systems," *ACM Trans. Graph.*, vol. 39, no. 5, pp. 155:1–155:13, Jun. 2020.
[52] A. Lindenmayer, *Adding Continuous Components to L-Systems*. Berlin, Germany: Springer, 1974, pp. 53–68.
[53] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Berlin, Germany: Springer-Verlag, 1996.
[54] R. Měch and P. Prusinkiewicz, "Visual models of plants interacting with their environment," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 397–410.
[55] P. Prusinkiewicz, M. James, and R. Měch, "Synthetic topiary," in *Proc. 21st Annu. Conf. Comput. Graph. Interactive Techn.*, 1994, pp. 351–358.
[56] T. Sakaguchi, "Botanical tree structure modeling based on real image set," in *Proc. ACM SIGGRAPH Conf. Abstr. Appl.*, 1998, Art. no. 272.
[57] I. Shlyakhter, M. Rozenoer, J. Dorsey, and S. Teller, "Reconstructing 3D tree models from instrumented photographs," *IEEE Comput. Graph. Appl.*, vol. 21, no. 3, pp. 53–61, May/Jun. 2001.
[58] G. Wang, H. Laga, J. Jia, N. Xie, and H. Tabia, "Statistical modeling of the 3D geometry and topology of botanical trees," *Comput. Graph. Forum*, vol. 37, pp. 185–198, 2018.
[59] G. Wang, H. Laga, N. Xie, J. Jia, and H. Tabia, "The shape space of 3D botanical tree models," *ACM Trans. Graph.*, vol. 37, no. 1, pp. 1–18, 2018.
[60] J. Binney and G. S. Sukhatme, "3D tree reconstruction from laser range data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 1321–1326.
[61] H. Huang and H. Mayer, "Extraction of 3D unfoliaged trees from image sequences via a generative statistical approach," in *Proc. 29th DAGM Conf. Pattern Recognit.*, 2007, pp. 385–394.
[62] A. Runions, B. Lane, and P. Prusinkiewicz, "Modeling trees with a space colonization algorithm," in *Proc. 3rd Eurographics Conf. Natural Phenomena*, 2007, pp. 63–70.
[63] B. Neubert, T. Franken, and O. Deussen, "Approximate image-based tree-modeling using particle flows," *ACM Trans. Graph.*, vol. 26, no. 99, 2007, Art. no. 88.
[64] C. Preuksakarn, F. Boudon, P. Ferraro, J.-B. Durand, E. Nikinmaa, and C. Godin, "Reconstructing plant architecture from 3D laser scanner data," in *Proc. 6th Int. Workshop Funct.- Struct. Plant Models*, 2010, pp. 12–17.
[65] K. Xie, F. Yan, A. Sharf, O. Deussen, H. Huang, and B. Chen, "Tree modeling with real tree-parts examples," *IEEE Trans. Visual. Comput. Graph.*, vol. 22, no. 12, pp. 2608–2618, Dec. 2016.
[66] S. Pirk *et al.*, "Plastic trees: Interactive self-adapting botanical tree models," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, 2012.
[67] S. Pirk *et al.*, "Modeling plant life in computer graphics," in *Proc. ACM SIGGRAPH Courses*, 2016, Art. no. 18.

[68] S. Pirk, T. Niese, O. Deussen, and B. Neubert, "Capturing and animating the morphogenesis of polygonal tree models," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–10, 2012.

[69] B. Benes and E. U. Millán, "Virtual climbing plants competing for space," in *Proc. Comput. Animation*, 2002, pp. 33–42.

[70] T. Hädrich, B. Benes, O. Deussen, and S. Pirk, "Interactive modeling and authoring of climbing plants," *Comput. Graph. Forum*, vol. 36, pp. 49–61, 2017.

[71] B. Wang, Y. Zhao, and J. Barbič, "Botanical materials based on biomechanics," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.

[72] S. Xu, F. Zhang, X. He, X. Shen, and X. Zhang, "PM-PM: PatchMatch with potts model for object segmentation and stereo matching," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2182–2196, Jul. 2015.

[73] W. T. Reeves, "Particle systems - A technique for modeling a class of fuzzy objects," *ACM Trans. Graph.*, vol. 2, no. 2, pp. 91–108, Apr. 1983.

[74] W. T. Reeves and R. Blau, "Approximate and probabilistic algorithms for shading and rendering structured particle systems," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 313–322, Jul. 1985.

[75] Y. Rodkaew, P. Chongstitvatana, S. Siripant, and C. Lursinsap, "Particle systems for plant modeling," in *Proc. Plant Growth Model. Appl.*, 2003, pp. 210–217.

[76] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," in *Proc. ACM SIGGRAPH Papers*, 2004, pp. 315–321.

[77] J. Long and M. Jones, "Reconstructing 3D tree models using motion capture and particle flow," *Int. J. Comput. Games Technol.*, vol. 2013, 2013, Art. no. 363160.

[78] X. Zhang, H. Li, M. Dai, W. Ma, and L. Quan, "Data-driven synthetic modeling of trees," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 9, pp. 1214–1226, Sep. 2014.

[79] T. Isokane, F. Okura, A. Ide, Y. Matsushita, and Y. Yagi, "Probabilistic plant modeling via multi-view image-to-image translation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2906–2915.

[80] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*. Cambridge, MA, USA: MIT Press, 1967, pp. 362–380.

[81] E. C. Sherbrooke, N. M. Patrikalakis, and F.-E. Wolter, "Differential and topological properties of medial axis transforms," *Graphical Models Image Process.*, vol. 58, no. 6, pp. 574–592, Nov. 1996.

[82] A. Bucksch, "A practical introduction to skeletons for the plant sciences," *Appl. Plant Sci.*, vol. 2, no. 8, 2014, Art. no. 1400005.

[83] P. K. Saha, G. Borgefors, and G. Sanniti di Baja, "A survey on skeletonization algorithms and their applications," *Pattern Recognit. Lett.*, vol. 76, pp. 3–12, Jun. 2016.

[84] A. Bucksch and R. Lindenbergh, "CAMPINO — A skeletonization method for point cloud processing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 63, no. 1, pp. 115–127, Jan. 2008.

[85] Z. Su, Y. Zhao, C. Zhao, X. Guo, and Z. Li, "Skeleton extraction for tree models," *Math. Comput. Modelling*, vol. 54, no. 3, pp. 1115–1120, Aug. 2011.

[86] J. Bartolozzi and M. Kuruc, "A hybrid approach to procedural tree skeletonization," in *Proc. ACM SIGGRAPH Talks*, 2017, Art. no. 53.

[87] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Trans. Graph.*, vol. 26, no. 4, pp. 19–es, Oct. 2007.

[88] A. Verroust and F. Lazarus, "Extracting skeletal curves from 3D scattered data," in *Proc. Shape Model. Int. Conf. Shape Model. Appl.*, 1999, pp. 194–201.

[89] S. Delagrange, C. Jauvin, and P. Rochon, "PypeTree: A tool for reconstructing tree perennial tissues from point clouds," *Sensors*, vol. 14, no. 3, pp. 4271–4289, Mar. 2014.

[90] D.-M. Yan, J. Wintz, B. Mourrain, W. Wang, F. Boudon, and C. Godin, "Efficient and robust reconstruction of botanical branching structure from laser scanned points," in *Proc. 11th IEEE Int. Conf. Comput.-Aided Des. Comput. Graph.*, 2009, pp. 572–575.

[91] C. Zhu, X. Zhang, M. Jaeger, and Y. Wang, "Cluster-based construction of tree crown from scanned data," in *Proc. Plant Growth Model. Simul. Vis. Appl.*, 2009, pp. 352–359.

[92] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," in *Proc. ACM SIGGRAPH Papers*, 2004, pp. 905–914.

[93] L. Gao, D. Zhang, N. Li, and L. Chen, "Force field driven skeleton extraction method for point cloud trees," *Earth Sci. Informat.*, vol. 12, no. 2, pp. 161–171, Jun. 2019.

[94] Y. Gong, Y. Yang, and X. Yang, "Three-dimensional reconstruction of the virtual plant branching structure based on terrestrial LiDAR technologies and l-system," in *Proc. Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, 2018, pp. 403–410.

[95] Y. Liu, J. Guo, B. Benes, O. Deussen, X. Zhang, and H. Huang, "TreePartNet: Neural decomposition of point clouds for 3D tree reconstruction," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 232:1–232:16, Dec. 2021.

[96] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[97] Y. Livny *et al.*, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Trans. Graph.*, vol. 29, no. 6, Dec. 2010, Art. no. 1.

[98] 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.

[99] 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.

[100] K. Kutulakos and S. Seitz, "A theory of shape by space carving," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 307–314.

[101] R. Spezialetti, D. J. Tan, A. Tonioni, K. Tateno, and F. Tombari, "A divide et impera approach for 3D shape reconstruction from multiple views," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 160–170.

[102] C.-H. Teng, Y.-S. Chen, and W.-H. Hsu, "Tree segmentation from an image," in *Proc. IAPR Conf. Mach. Vis. Appl.*, 2005, pp. 59–63.

[103] J. Zeng, Y. Zhang, and S. Zhan, "3D tree models reconstruction from a single image," in *Proc. - 6th Int. Conf. Intell. Syst. Des. Appl.*, 2006, pp. 445–450.

[104] P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan, "Single image tree modeling," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 108:1–108:7, Dec. 2008.

[105] J. Guénard, G. Morin, F. Boudon, and V. Charvillat, "Reconstructing plants in 3D from a single image using analysis-by-synthesis," in *Proc. Int. Symp. Vis. Comput.*, 2013, pp. 322–332.

[106] J. Guenard, G. Morin, F. Boudon, and V. Charvillat, "Realistic plant modeling from images based on analysis-by-synthesis," in *Proc. Int. Conf. Math. Methods Curves Surfaces*, 2014, pp. 213–229.

[107] Z. Liu, K. Wu, J. Guo, Y. Wang, O. Deussen, and Z. Cheng, "Single image tree reconstruction via adversarial network," *Graphical Models*, vol. 117, Sep. 2021, Art. no. 101115.

[108] C.-H. Teng and Y.-S. Chen, "Image-based tree modeling from a few images with very narrow viewing range," *Vis. Comput.*, vol. 25, no. 4, pp. 297–307, Apr. 2009.

[109] L. D. Lopez, Y. Ding, and J. Yu, "Modeling complex unfoliaged trees from a sparse set of images," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2075–2082, 2010.

[110] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 92, pp. 79–97, 2014.

[111] Y.-H. Tu, S. Phinn, K. Johansen, A. Robson, and D. Wu, "Optimising drone flight planning for measuring horticultural tree crop structure," *ISPRS J. Photogrammetry Remote Sens.*, vol. 160, pp. 83–96, 2020.

[112] J. Qi, J. Gastellu-Etchegorry, and T. Yin, "Reconstruction of 3D forest mock-ups from airborne LiDAR data for multispectral image simulation using dart model," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 3975–3978.

[113] W. Xiangyu, X. Donghui, Y. Guangjian, Z. Wuming, W. Yan, and C. Yiming, "3D reconstruction of a single tree from terrestrial LiDAR data," in *Proc. IEEE Geosci. Remote Sens. Symp.*, 2014, pp. 796–799.

[114] K. Kuželka and P. Surový, "Mathematically optimized trajectory for terrestrial close-range photogrammetric 3D reconstruction of forest stands," *ISPRS J. Photogrammetry Remote Sens.*, vol. 178, pp. 259–281, 2021.

[115] J. Shao *et al.*, "SLAM-aided forest plot mapping combining terrestrial and mobile laser scanning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 163, pp. 214–230, 2020.

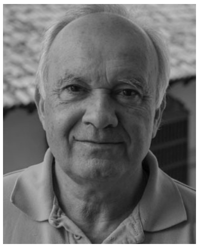
[116] Z. Su, S. Li, H. Liu, and Z. He, "Tree skeleton extraction from laser scanned points," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 6091–6094.

[117] H. Huang, L. Tang, and C. Chen, "A 3D individual tree modeling technique based on terrestrial LiDAR point cloud data," in *Proc. 2nd IEEE Int. Conf. Spatial Data Mining Geographical Knowl. Serv.*, 2015, pp. 152–156.

[118] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: Int. J. Geographic Inf. Geovisualization*, vol. 10, no. 2, pp. 112–122, Dec. 1973.
[119] F. Boudon, A. Meyer, and C. Godin, "Survey on computer representations of trees for realistic and efficient rendering," LIRIS UMR CNRS5205, Res. Rep. 2301, 2006.
[120] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. Kang, "Image-based plant modeling," *ACM Trans. Graph.*, vol. 25, pp. 599–604, 2006.
[121] D. Bradley, D. Nowrouzezahrai, and P. Beardsley, "Image-based reconstruction and synthesis of dense foliage," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 74:1–74:10, Jul. 2013.
[122] M. Pollefeys and L. V. Gool, "From images to 3D models," *Commun. ACM*, vol. 45, no. 7, pp. 50–55, 2002.
[123] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Comput. Graph. Forum*, vol. 32, no. 5, pp. 113–123, 2013.
[124] H. Huang, L. Tang, and C. Chen, "A 3D individual tree modeling technique based on terrestrial LiDAR point cloud data," in *Proc. 2nd IEEE Int. Conf. Spatial Data Mining Geographical Knowl. Serv.*, 2015, pp. 152–156.
[125] Z. Su, S. Li, H. Liu, and Z. He, "Tree skeleton extraction from laser scanned points," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 6091–6094.
[126] O. Argudo, C. Andújar, and A. Chica, "Image-based tree variations," *Comput. Graph. Forum*, vol. 39, no. 1, pp. 174–184, 2020.
[127] B. Li *et al.*, "Learning to reconstruct botanical trees from single images," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 231:1–231:15, Dec. 2021.
[128] C. Li, O. Deussen, Y.-Z. Song, P. Willis, and P. Hall, "Modeling and generating moving trees from video," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 1–12, 2011.
[129] Y. Lin, J. Liu, and J. Zhou, "A novel tree-structured point cloud dataset for skeletonization algorithm evaluation," CoRR, 2020.
[130] A. López, C. J. Ogayar, J. M. Jurado, and F. R. Feito, "A GPU-accelerated framework for simulating LiDAR scanning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, Apr. 2022.
[131] A. Graciano, A. J. Rueda-Ruiz, A. Pospisil, J. Bittner, and B. Benes, "QuadStack: An efficient representation anddirect rendering of layered datasets," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 9, pp. 3733–3744, Sep. 2021.
[132] S. Kolmanič, D. Strnad, Š. Kohek, B. Benes, P. Hirst, and B. Žalik, "An algorithm for automatic dormant tree pruning," *Appl. Soft Comput.*, vol. 99, 2021, Art. no. 106931.
[133] L. Bosheng *et al.* ., "Learning to reconstruct botanical trees from single images," in *Proc. SIGGRAPH ASIA*, 2021, Art. no. 231.

José-L. Cárdenas received the MS degree in computer science from the University of Jaén, in 2019, where he is currently working toward the PhD degree in the Computer Graphics and Geomatics Group. His research interests include 3D modeling and rendering.

Carlos J. Ogayar received the MS degree in computer science and the PhD degree in computer science from the University of Granada, in 2001 and 2006, respectively. He is currently an associate professor of the Department of Computer Science, University of Jaén (Spain). His research focus is on GPU computing, geometric algorithms, virtual reality and 3D scanned data processing.



Francisco R. Feito received the BSc degree in mathematics from the University Complutense of Madrid and the PhD degree in computer science from the University of Granada, Spain. He is currently a full professor with the Department of Computer Science, University of Jaén. He has been teaching with the High Polytechnics Institute, Jaén, and the head of the Department and the Graphics and Geomatics Research Group. His research interests include formal methods for computer graphics, geometric modeling, computational geometry, geographical information sciences, digital twins, virtual archaeology, and precision agriculture.

Juan M. Jurado received the MS degree in computer science, in 2017, and the award of the best master's thesis of the University of Jaén, the MS degree in high performance computing from the University of A Coruña, in 2020, and the PhD degree in computer science from the University of Jaén. He is currently an assistant professor of the Department of Software Engineering, University of Granada, Spain. His research focuses on the generation and processing of 3D models, multi-source data fusion, as well as the monitoring, modeling and analysis of biophysical features in 3D real-world environments.