Realistic Procedural Plant Modeling from Multiple View Images

Jianwei Guo, Shibiao Xu, Dong-Ming Yan, Zhanglin Cheng, Marc Jaeger, Xiaopeng Zhang

Abstract—In this paper, we describe a novel procedural modeling technique for generating realistic plant models from multi-view photographs. The realism is enhanced via visual and spatial information acquired from images. In contrast to previous approaches that heavily rely on user interaction to segment plants or recover branches in images, our method automatically estimates an accurate depth map of each image and extracts a 3D dense point cloud by exploiting an efficient stereophotogrammetry approach. Taking this point cloud as a soft constraint, we fit a parametric plant representation to simulate the plant growth progress. In this way, we are able to synthesize parametric plant models from real data provided by photos and 3D point clouds. We demonstrate the robustness of the proposed approach by modeling various plants with complex branching structures and significant self-occlusions. We also demonstrate that the proposed framework can be used to reconstruct ground-covering plants, such as bushes and shrubs which have been given little attention in the literature. The effectiveness of our approach is validated by visually and quantitatively comparing with the state-of-the-art approaches.

1 Introduction

Plants are ubiquitous in the nature, and realistic plant modeling plays an important role in various applications, which range from forest management and urban modeling to virtual reality, such as 3D games and scene roaming. Plant reconstruction and modeling have been fundamental research topics for a long time in the fields of computer graphics [1], remote sensing [2], and plant biology [3].

A large amount of plant modeling approaches have been developed in the past decades, however, plant modeling remains a non-trivial and challenging task due to the complexity of the modeling process. For example, procedural modeling approaches [4], [5], [6] work efficiently for synthesizing local branching structure details to produce botanically correct plant models. Although this conventional approach has certain capability to control the growth of plants under certain shape constraints [7], it is quite difficult to model existing real-world plants. Recently, reconstruction from point clouds [8], [9], [10] has received considerable attention. While these geometry-based methods might precisely reconstruct skeletal structures, the botanical fidelity of plants is difficult to maintain. Another drawback is that 3D laser/LIDAR scanning often involves high device cost, inconvenient operation, and sensitivity to occlusion, which greatly limit its practical application in many complex scenes.

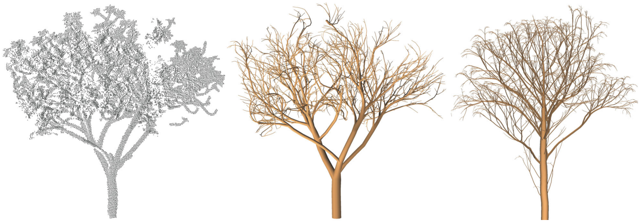


Fig. 1. Modeling results using our approach (Middle) and the method of Pałubicki et al. [14] (Right). The parameters used in both approaches are same.

With the fine portability and high resolution of optical cameras, image-based approaches provide a cost-effective way that overcomes the above mentioned problems. The structure-from-motion (SFM) method is typically used to reconstruct the point clouds from multiple plant images [11] or even fewer images with narrow viewing ranges [12]. However, the state-of-the-art SFM methods still have serious limitations. The reason is threefold: first, since the feature-matching process based on features [e.g., scale-invariant feature transform (SIFT)] does not perform efficiently on plants with complexity and self-hiding properties, these methods produce unsatisfying point cloud that may cause errors for reconstruction; second, user interactions are usually required to identify the crown or branching structures on images to avoid segmentation, which can be difficult and tedious; last but not least, these approaches tend to accomplish a reconstruction mission rather than a modeling issue, and thus, the output is difficult to use in later applications, such as dynamic reaction to the environment [13].

We present a new framework for generating realistic plants in this paper to fully possess advantages of

- *J. Guo, S. Xu, D.-M. Yan, and X. Zhang are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.*
- *Z. Cheng is with Shenzhen VisuCA Key Lab, SIAT, CAS, China.*
- *M. Jaeger is with LIRMM-ICAR, Montpellier University, France.*

procedural modeling and data-driven reconstruction approaches. We connect both data-driven reconstruction and plant growth modeling for our procedural modeling of plants. The technique of plant growing is used in our approach for two reasons. One is that the procedural modeling is closely growth related; hence, the procedural information is recovered through the growth of plant structure. The other is that our plant reconstruction needs an expansion of the tree structure which starts from the root position of the target plant according to the shape constraint. In this sense, "growth" means space expansion of tree branches to fill to the space of the input data, and we mention about plant growth in our technique for structure recovery. Therefore, our results are mainly plant models that cannot be directly used for growth simulation.

In this paper, we make several novel improvements over existing image-based modeling approaches. First, we extend the binocular stereo approach to multiple views to generate depth maps for each view; thus, a dense point cloud is obtained by projecting and fusing all depth information together. Second, we present a rule-based framework for generating a realistic plant model from a point cloud, which avoids finding parameters required in optimization-based methods [15] [16]. In this way, we integrate multi-view point cloud acquisition with rule-based approach for procedural plant modeling. This combination enriches the plant generation literature by building connections between rule-based virtual plant modeling and data-driven plant reconstruction. In summary, our method supplements several important contributions to plant modeling:

1) We develop an automatic 3D point cloud acquisition technique for plants via accurate and dense depth map computation for plant images, which could generate denser and more complete point cloud than those generated by traditional multi-view stereo reconstruction approaches.
2) We present a new plant reconstruction method by combining real data (photos and 3D point cloud) analysis with rule-based procedural plant modeling. Specifically, we could reconstruct a realistic plant from the point cloud using a procedural model, as shown in Fig. 1.
3) The proposed framework can also be used to model ground-covering non-tree plants, such as bushes and shrubs, which have gained little attention and cannot be handled in previous reconstruction based approaches [9], [10], [17].

2 Related Work

The plant modeling approaches can be classified into three major categories, namely, procedural modeling, data-driven reconstruction approaches (e.g., photographs or scanned points), and interactive modeling. A detailed survey is out of the scope of this paper, and we briefly review some key techniques in this section. In addition, the most relevant works for computing dense depth maps are also revisited.

Virtual plant modeling. Rule-based procedural modeling approaches utilize a small set of generative rules deduced from plants that occur in the nature to create complex branching structures. The most famous rule-based system is the L-system [4]. Prusinkiewicz et al. develop a series of algorithms [18], [19], [20] based on L-system to model different plant types with special characteristics. Honda [5] considers constructing the tree structure through a recursive procedure under some rules by using a small number of geometric parameters. Since then, numerous geometric rules, such as implicit functions [21], fractal models [22], and narrow near-conical tubes [23], are used to produce recursive realistic-looking plants. De Reffye et al. [24] also use a collection of rules, but these rules are motivated by plant growth models. Deussen and Lintermann [1] develop the Xfrog technique as a combination of a rule-based technique and geometric modeling. More recently, the space colonization algorithms [6], [25] is proposed to generate trees and shrubs by simulating the competition for space among growing branches. Pałubicki et al. [14] generate realistic models of trees and shrubs from a self-organizing process dominated by the competition of buds and branches for light and space. Kim and Cho [26] introduce growth volume for efficient modeling of trees by means of botany-based self-organizing under a recursive hierarchy structure.

Instead of directly modeling plants from generative rules, other approaches use guidance shapes as constraints. Stava et al. [16] propose an inverse procedural method to guide L-system-based plants toward a target polygonal model. This method uses Monte Carlo Markov Chains (MCMC) to estimate the optimal parameters of a procedural model for producing plants similar to the input. Wang et al. [15] propose a new variational approach targeting on generating realistic plants in specific shapes, in which the shape-guidance plant modeling is formulated as an optimization problem.

In other works, procedural models can be transformed into certain shapes by user input. For example, Beneš et al. [27] present a guided procedural modeling approach that divides space into guide regions for branch growth, which allows the user to control the high-level description by editing the guides. User-drawn sketches have also been used for interactive design and editing of 3D plants [28]. Xu and Mould [29], [30] present graph-based methods to compute a branching structure by collecting least-cost paths in a graph; users can create desired variations by adjusting the initial graph shape.

Data-driven reconstruction of plants. Nowadays, the approaches of reconstructing plants from real-world data have been proposed, which reflect the real shape of a plant with high precision and high resolution.

Many researchers have attempted to utilize 3D reconstruction techniques from 2D photos or videos [31] for plant modeling. The pioneering work of Shlyakhter et al. [32] extracts visual hulls from input images and constructs a medial axis as the main 3D tree skeleton. The branches are synthesized within these hulls using L-systems. Reche-Martinez et al. [33] present a volumetric representation for the rendering of plants with variable opacity. However, this method lacks the real 3D geometry of plants because it is based on billboards. The method of Quan et al. [34] can produce an accurate leaf geometry using SFM techniques, but it hardly scales to outdoor and large plants. Tan et al. [17] automatically synthesize L-system rules from input images to accommodate the modeling of large-sized plants, which complements the approach of [34] that is only suitable for small-sized plants. In a follow-up work, Tan et al. [35] generate a realistic 3D plant model from only a single image, but they rely on user-drawn strokes to guide the synthesis. Neubert et al. [36] generate an approximate volumetric plant representation, and a 3D particle flow simulation is performed to extract branching structure from the recovered plant volume. Bradley et al. [37] introduce a framework for 3D capture, modeling, and synthesis of densely leaved foliage, which is based on multi-view stereo combined with data-driven tactics. Guénard et al. [38] propose to generate a 3D plant model using an analysis-by-synthesis method that combines information from a single image and a priori knowledge of the plant species.

In contrast to image-based methods, 3D point cloud obtained via laser scanning provides a more intuitive way to capture plant geometry [39], [40]. Xu et al. [8] classify points into leaves and branches, and they develop a heuristic-based approach to model main branches by clustering edges in a spanning graph. Small twigs and leaves are randomly added to form the crown geometry. Later, a hybrid approach [41] extracts 3D skeletons also using a spanning tree-based algorithm. Bucksch et al. [42], [43] apply space partitioning to cluster points and build a curved skeleton by connecting adjacent clusters. Côté et al. [44] synthesize plant skeleton and leaf geometry based on light scattering properties obtained from scanned sample intensities. Yan et al. [45] reconstruct complete branches by fitting cylinders on local parts of a segmented point cloud. Livny et al. [9] use global optimizations for reconstructing tree skeletons by computing minimal spanning graphs. In a subsequent work, they present a lobe-based representation and synthesize a full plant model by instancing the lobes with predefined patches [46]. Raumonen et al. [47] locally approximate trunks and branches with cylinders which are then combined into branches. Friedman and Stamos [48] infer shape grammars from the wavelet transform of an input tree point cloud then reproduce the tree structure using the grammars. Li et al. [49] present a forward-backward analysis approach for capturing a developing plant and analyzing its evolving parts over time. Wang et al. [10], [50] present structure and direction-aware global optimization approaches to reconstruct 3D plants from incomplete point clouds. Recently, the work of [51] proposes a data-driven approach to synthesize 3D botanical trees from existing ones using a statistical modeling in which each tree is considered a point in a tree-shaped space, equipped with a proper metric.

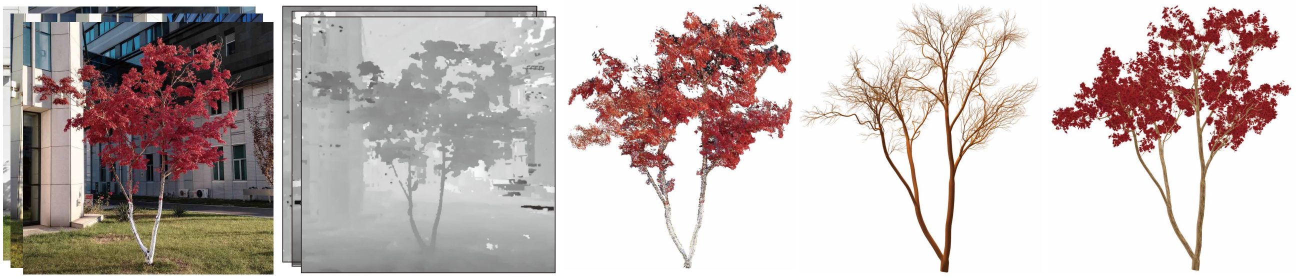
Multi-view stereo reconstruction. The state-of-the-art point cloud reconstruction techniques include two steps, namely, SFM [52], [53] and Patch-based Multi-View Stereo (PMVS) [54]. This combination has been widely used for multiple plant images [11], [17], [34]. SFM can generate a sparse set of matched key points (e.g., SIFT), and PMVS repeatedly expands these features to semi-dense point clouds based on visibility constraints. However, these methods aim at reconstructing a global 3D model by using all the images available simultaneously; thus, they suffer from the scalability problem as the number of images increases. Their feature matching process also may not perform well on plant images with repetitive or similar feature property, which cannot produce point clouds to satisfy our plant modeling requirements.

By contrast, we pay attention to depth map based reconstruction methods, which are natural extensions from binocular stereo to multiple views. Such methods first compute dense depth maps at each view and then project them together into a single point cloud model by considering visibility. Shen [55] uses PatchMatch stereo [56] to generate accurate depth maps instead of merely matching key point features. In this paper, we further combine the PatchMatch stereo algorithm with image segmentation to improve the initial depth estimation for repetitive or similar feature regions. We also enable a global bundle optimization for depth refinement according to the geometric coherence constraint, which is solved via graph cuts [57] to take advantage of better convergence for achieving high accuracy.

3 Methodology Overview

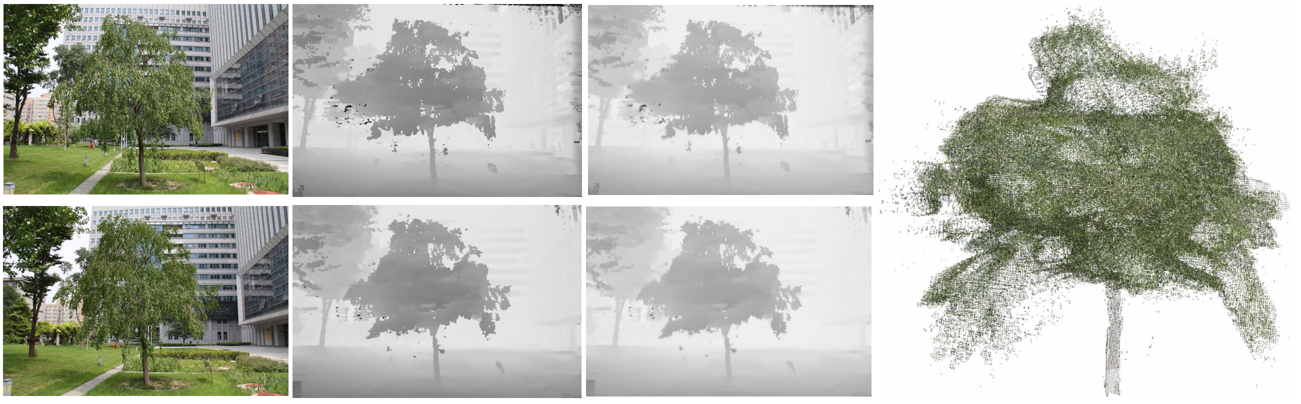
Our goal is to generate a realistic plant model from a handful of overlapping images. Fig. 2 illustrates the overall plant modeling process, which comprises two main steps: 3D point cloud reconstruction and procedural modeling guided by point cloud.

Initially, we develop an effective method to compute dense depth map for each image. After recovering camera parameters, we consider the photo-consistency and geometric coherence constraints over multiple views to estimate per-pixel depth (Fig. 2(b)), which is then refined by an iterative global optimization. The depth maps can be easily combined into



(a) Input images (b) Depth maps (c) Point cloud (d) Skeletal structure (e) Textured plant

Fig. 2. Overview of our image-based plant modeling system. Given a set of multi-view images (a), we first compute a dense depth map for each image (b), and extract a single point cloud from the depth maps (c). Guided by the point cloud, a rule-based modeling technique is then applied to reconstruct the plant skeleton (d). (e) shows the final plant model with textured leaves.



(a) (b) (c) (d)

Fig. 3. Key steps of extracting point cloud process. From left to right: input images (a), per-pixel depth estimation by considering the photo-consistency and geometric coherence constraints (b), depth refinement by guided filtering (c), reconstructed point cloud (d). Here we show two views for the depth map computation.

a single dense point cloud by considering visibility (Fig. 2(c)).

Next, we regard the point cloud as soft constraints to guide the procedural plant modeling. In order to describe various plant species, a parametric skeletal structure is presented, where the parameters are stored in a data repository. We integrate this parametric representation into a rule-based growing system by adapting this system to the real point cloud (Fig. 2(d)). Finally, the 3D skeletal structure is converted into plant geometry by meshing branches and adding leaves to complete the model, as shown in Fig. 2(e).

4 Point Cloud Reconstruction

Given an image sequence, camera parameters can be estimated reliably by employing the open-source Bundler SFM system [53], [58]. To make accurate depth estimation, we introduce an iterative optimization scheme by first initializing the depth maps based on photo-consistency and then refining the depth by geometric coherence.

Depth initialization. We define the camera parameters of two neighboring images I_i, I_j as $\{K_i, R_i, C_i\}$ and $\{K_j, R_j, C_j\}$, where K is the intrinsic matrix, R is the rotation matrix, and C is the camera center. The transformation matrix [59] between I_i and I_j is given by

$$H_{i\to j}(d_{\mathbf{x}_i}, \mathbf{n}_i) = K_j(R_j R_i^{-1} - R_j(C_j - C_i)\mathbf{n}_i^T/v)K_i^{-1}. \tag{1}$$

where parameter $v = -\mathbf{n}^T X$ is defined via normal $\mathbf{n}$ and coordinate X of the current 3D point.

For each pixel $\mathbf{p}$ in image I_i, we can find its corresponding pixel in I_j using this transformation matrix $H(d, \mathbf{n})$. For applying the PatchMatch stereo, we combine the mean-shift segmentation and photo-consistency to define the depth estimation problem as:

$$\mathcal{E}_1(d, \mathbf{n}) = \tau \int_{\Omega_i} m(\mathbf{p}, (d, \mathbf{n}))d\mathbf{p} + \lambda \int_{\Omega_i} \|d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p})\|_1 d\mathbf{p}, \tag{2}$$

where we model each segment Ω_i as a 3D plane. $m(\mathbf{p}, (d, \mathbf{n}))$ is the photo-consistency based matching cost for pixel $\mathbf{p}$ [59], while $\|d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p})\|_1$ represents

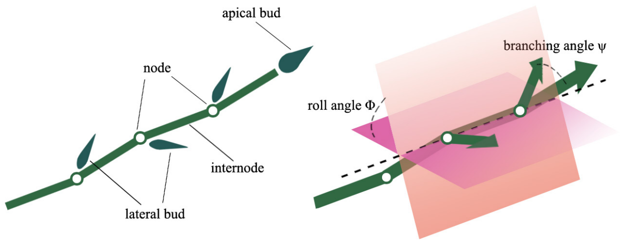


Fig. 4. Illustration of terminologies (left) and parameters (right) used in our parametric plant representation.

the depth cost in each segment, which enforces that a segment has a compact set of depth values.

Depth refinement. Since the raw depth maps may not completely agree with each other on overlapping areas due to depth errors, a refinement process is needed to enforce geometric coherence over multiple views. We accordingly give the likelihood of the depth for any pixel $\mathbf{p} \in I_i$ by combining the photo-consistency and geometric coherence constraints:

$$L(\mathbf{p},(d_i,\mathbf{n}_i)) = \sum_{\mathbf{i}'} m(\mathbf{p},(d_i,\mathbf{n}_i))m_v(\mathbf{p},\mathbf{p}',(d_{i'},\mathbf{n}_{i'})), \tag{3}$$

where the geometric coherence term $m_v(\mathbf{p},\mathbf{p}',(d_{i'},\mathbf{n}_{i'}))$ is similar to the constraint used in [60]. Our optimization problem is finally defined as:

$$\mathcal{E}_2(\mathbf{p},(d,\mathbf{n})) = \int_{I_i} (1-u(\mathbf{p})L(\mathbf{p},(d_i,\mathbf{n}_i)))d\mathbf{p}, \tag{4}$$

where $u(\mathbf{p})$ is an indicator function. Fig. 3(b) shows the depth results by considering both the photo-consistency and geometric coherence constraints.

For an efficient and accurate optimization, we use the standard graph cuts approach [57] to minimize the above energy function. Further, it is worth noting that a high quality depth map of plants should have smooth and detailed structures. Hence, we apply a guided filtering algorithm [61] to further refine the depth map. The refined depth map manages to capture the detailed edge discontinuities and outline the profile of plants, as shown in Fig. 3(c).

5 Plant Modeling

Once the 3D point cloud has been extracted, we generate a realistic plant model. Unlike previous approaches [9], [17], which focus on accomplishing a reconstruction mission, we fit a parametric model to the points of plants by applying a rule-based modeling method. In the following, we first describe a parametric representation for plant structure, which can define various geometric plant shapes. Then, an improved procedural modeling approach guided by real point cloud is proposed.

TABLE 1
Tree parameters.

Parames.	Name	Description
l	internode length	the base length of a single internode
ϕ	roll angle	rotation angle of lateral branches associated with two successive nodes
ψ	branching angle	angle between a lateral branch and its parent shoot
ρ	growth units	number of internodes generated on a single shoot during one growth cycle
γ	diameter coefficient	branch thickness transmission coefficient

5.1 Parametric plant representation

In this section, we build a parametric representation for plant skeletal structure, as shown in Fig. 4. To precisely explain our techniques, we first give the definition of several related terminologies.

Terminologies. In a hierarchically organized plant structure [24] [14], a *node* is a basic concept that corresponds to a point composing a branch and probably supporting stems or leaves. The part of a branch between two successive nodes is called *internode*. There are two kinds of buds, namely, a *lateral bud* is created in the axil of each leaf, and an *apical bud* is situated at the end of a branch. The fundamental structural unit of a plant body is known as *metamer* formed by a node, associated with its leaf (or leaves) and lateral bud(s) plus the subtending internode. A sequence of metamers created from active buds forms a *shoot*.

Parametric skeletal structure. We create a parametric skeletal structure that is sufficiently powerful to generate various plant species. Several procedural modeling methods exist for extracting branching structures, and theoretically, any parameter-driven approaches can be integrated into our framework. For example, Stava et al. [16] use 24 parameters to simulate the effects of internal and external factors on plant shape. However, since our modeling approach is guided by real data, many parameters in [16] is redundant for us, such as the environmental and bud fate parameters. On the other hand, tuning so many parameters is difficult for ordinary users. Therefore, in our work, we employ a parametric model that includes 5 parameters for representing the skeletal structure. These parameters are listed in Table 1, and illustrated in Fig. 4. We pre-generate those parameters by measuring multiple real trees and store the average value in a species library. The parameter of growth units ρ can be computed automatically in our method by using a logistic growth equation, which will be described later in subsection 5.2.

The last parameter γ is used for computing branch thickness at every growing stage. From the reconstructed point cloud, we can estimate the diameter of the trunk at the plant base d_{root}, then the allometric rule [62] discovered by Da Vinci is used to obtain plausible estimates for the rest of the plant skeleton. Given the diameter d_p of the parent node, the diameter of each child node is defined by Equation 5:

$$d_{c_i} = d_p\left(\frac{l_{c_i}}{\sum_j l_{c_j}}\right)^{\gamma}, \tag{5}$$

where l_{c_i} is the total branch length supported by this child.

5.2 Procedural modeling

The core of our modeling algorithm is built on the observation that an obtained point cloud represents a plant growing result controlled by external environmental and internal factors. The original space colonization algorithm [6] distributes markers (uniformly and randomly generated points) to guide the growth process, where the markers represent the available space. However, this algorithm is mostly used to model virtual plants without controlling the form of plants. In order to control the tree form, some interactive systems, such as the works of Pałubicki et al. [14] and TreeSketch [28], use a 3D procedural brush to dynamically create markers. Inspired by this idea, we assume that the geometry of plants has been formed by the point cloud (or depth information), just imagine we have pre-generated such points using a 3D procedural brush. Then, starting from seeds located at the root, we improve a parametric procedural model to simulate plant modeling, i.e., buds always grow toward some points of the point cloud and fit those points.

Rule-based growing mechanism. In this step, we integrate the parametric representation into a rule-based growing mechanism using a relatively simple L-system, which describes branching plants by string-rewriting mechanisms. From a seed location, our L-system is defined as follows:

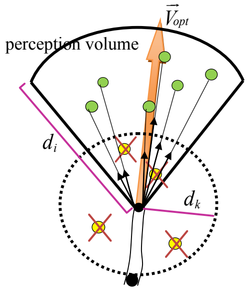
L-system 1: Simulate plant growing

```
$Seed(\mathbf{p},\mathbf{v}) \longrightarrow A(\mathbf{p},\mathbf{v})$
$A(\mathbf{p},\mathbf{v}) \longrightarrow \{Metamer(\mathbf{p}',\mathbf{v}')*\}A(\mathbf{p}'',\mathbf{v}'')$
$Metamer(\mathbf{p},\mathbf{v}) \longrightarrow Internode(\mathbf{p},\mathbf{v})L(\mathbf{p}',\mathbf{v}')$
$L(\mathbf{p},\mathbf{v}) \longrightarrow A(\mathbf{p}',\mathbf{v}')$
```

Starting from a seed with position $\mathbf{p}$ and orientation $\mathbf{v}$, we set the seed as the apical bud of the future trunk. Here, orientation $\mathbf{v}$ indicates the growth direction of each bud. Its initial value is the upward direction, but will be adjusted according to the branching and roll angles at each stage (see the following). The developmental rule (the kernel of the growth process) is defined from the apical bud $A(\mathbf{p},\mathbf{v})$. This rule results in a chain of metamers (a growth unit, or growth shoot), represented by $\{*\}$. For each metamer, the position is computed from the previous one (the closest metamer on the same branch in the growth direction), using the previous orientation and length. The orientation is also updated by applying the roll (phyllotaxy) angle. For example, n is supposed to be the total number of metamers generated in one cycle ($n \geq 1$). For the first metamer ($i = 1$), $\mathbf{p}'_1 = \mathbf{p}, \mathbf{v}'_1 = \mathbf{v}$, where the new metamer position is the apical bud position, same for the direction. For the next metamers ($i > 1, i \leq n$), the new metamer position is at the end of the previous one: $\mathbf{p}'_i = \mathbf{p}'_{i-1} + \mathbf{v}'_{i-1} \cdot l$, and the new direction is a phyllotaxy application to the previous metamer: $\mathbf{v}'_i = R_\phi(\mathbf{v}'_{i-1})$, where R_ϕ represents the rotation matrix specified by the roll angle ϕ. Note that this rotation does not change the metamer main direction but applies the phyllotaxy to the lateral bud. Finally, the apical bud closes the metamer chain: $\mathbf{p}'' = \mathbf{p}'_n + \mathbf{v}'_n \cdot l$, $\mathbf{v}'' = \mathbf{v}'_n$.

Then in concordance with botany plant architecture, the metamer is interpreted as an internode and a lateral bud $L(\mathbf{p}',\mathbf{v}')$, where the position of L is at the end of the internode, and the orientation of the lateral L is transformed with respect to the branching angle parameter ψ: $\mathbf{p}' = \mathbf{p} + \mathbf{v} \cdot l$, $\mathbf{v}' = R_\psi(\mathbf{v})$. Finally, any lateral bud is considered a new apical bud for a new axis to be grown in the next growth cycle.

Space colonization. When implementing the above L-systems, Pałubicki et al. [14] iteratively simulate the space competition among growing branches. They assume that each bud is surrounded by a spherical occupancy zone of radius d_k (called kill distance) and has a conical perception volume characterized by the perception angle θ and radius d_i (called influence distance), as shown in the inset below. Typical values for these parameters are: $\theta \approx 90°$, $d_i \in [4l, 12l]$, and $d_k = 0.5d_i$, where l is the internode length. Then each bud searches its perception volume to find the markers that are closest to it. Finally, the optimal direction, $\vec{V}_{opt}$, of its growth is calculated as the average of the normalized vectors $\vec{V}_i$ formed by the bud and each marker: $\vec{V}_{opt} = \sum_{i=1}^{n} \vec{V}_i$. The growing distance is the distance from the bud to the centroid of these markers.



While this framework can generate plausible results, it has not taken real data into account. As a result, it is not easy to generate plants that are similar to real data (see Fig. 1). In our work, we integrate three adaptation steps into this method. First, we assume that the geometry of plants has been formed by the point cloud. Thus, we consider the point cloud as the markers of the space within which we should produce branches. Second, in the reconstructed point cloud, the markers around each bud are not uniformly distributed, and different parts of the plant

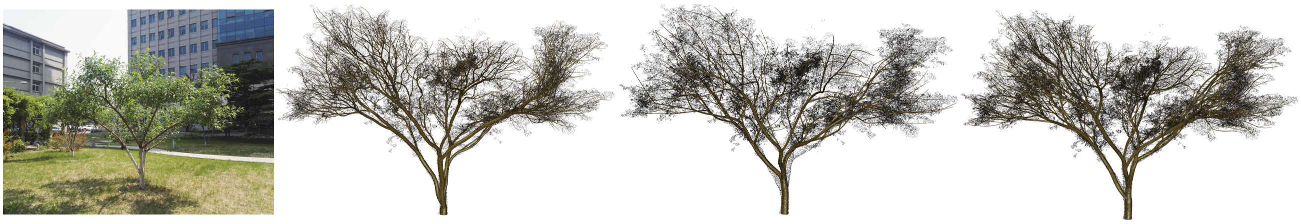


Fig. 5. Effect of our logistic growth model. From left to right: a reference photo, modeling result using a smaller constant influence distance ($5l$), modeling result using a larger constant influence distance ($10l$), and modeling result using the influence distance adaptively computed from the logistic growth equation.

point cloud have different influences on the skeleton growth. Third, the influence distance should not equal at every stage in real plant growing process. For example, in early stages, plant grows fast because the space and light resources are sufficient for each bud; in later stages, plant grows slowly. In the following, we describe these improvements for this method to adapt to the reconstructed point cloud.

Point cloud guided modeling. We present an automatic method to adaptively compute the influence distance d_i at each stage. Previous studies [63], [64] by botanists have shown that plants grow in such a way that the growth increases rapidly when the plants are young and decreases as the plants approach their asymptotic maximum growth capacity. As a result, branch diameter and length are mainly related to the age of each branch. To simulate this nonlinear plant growth, we apply the well-known Logistic model that describes the balance between branch growth and competition. Given the internode length l, we let $H_{max} = K \cdot l$ be the maximal branch length from the leaf to the root, where K is the maximum of internode quantities from leaf nodes to the root. We assume that the plant height approximates H_{max}. Therefore, our logistic growth equation can be defined as:

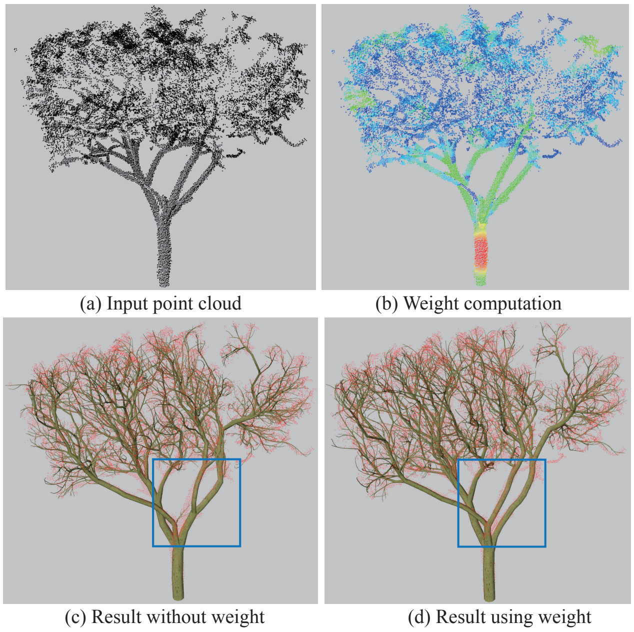
$$H_t = \frac{H_{max}}{1 + (\frac{H_{max}}{H_0} - 1)e^{-\rho_0 t}}, \tag{6}$$

where t is the growth iteration time, H_t is the maximal branch length at time t with default initial value $H_0 = 2$, $\rho_0 = 0.3$ is the initial growth units. Then the growth units equation is obtained by the derivation of equation (6):

$$\rho_t = \frac{dH}{dt} = \frac{H_{max} \cdot \rho_0 \cdot (\frac{H_{max}}{H_0} - 1)e^{-\rho_0 t}}{(1 + (\frac{H_{max}}{H_0} - 1)e^{-\rho_0 t})^2}. \tag{7}$$

Then ρ_t determines the influence distance $d_i = \rho_t \cdot l$ at each stage. In Fig. 5, we show the influence of the logistic growth equation. By using this equation, we can adaptively adjust the influence distance, thus causing the parametric plant to fit to the point cloud.

Next, in the work of [14], they treat every point in perception volume equally, i.e., each point has the same weight for skeleton computation. However, we find that the points on branches usually exert larger influences on the skeleton growth than on leaf points. Therefore, in our approach, each point is equipped with two attributes, namely, location and importance weight. To compute the weight for each point $\mathbf{p}_i$, we first perform *Principal Component Analysis* (PCA), and store the eigen vector, $\vec{\boldsymbol{n}}_{\mathbf{p}_i}$, with the maximum eigen value for $\mathbf{p}_i$. We search in the neighborhood of $\mathbf{p}_i$ and count the number of points $\mathbf{p}_j$ that have similar location and eigen vector to those of $\mathbf{p}_i$:



(a) Input point cloud

(b) Weight computation

(c) Result without weight

(d) Result using weight

Fig. 6. Example that shows our weight with color coding (warm color indicates large weight), as well as the modeling results without/with the weight.

$$\|\mathbf{p}_i - \mathbf{p}_j\|_2 < \varepsilon_D, (\vec{\boldsymbol{n}}_{\mathbf{p}_i} \star \vec{\boldsymbol{n}}_{\mathbf{p}_j}) > \varepsilon_A, \tag{8}$$

where $\|\cdot\|_2$ denotes the L_2 Euclidean norm and $\star$ is the dot product between two unit normal vectors. We set these default thresholds to $\varepsilon_D = 3cm$ and $\varepsilon_A = 0.9$. The number of similar neighbors is denoted as weight w_i, which can be considered the density of $\mathbf{p}_i$ determined by its location and principal direction. Finally, the actual growth direction of each bud $\mathcal{A}$ is computed as follows:

$$\vec{\boldsymbol{V}}_{\mathcal{A}} = \vec{\boldsymbol{V}}_{default} + \alpha\vec{\boldsymbol{V}}_{opt} + \beta\vec{\boldsymbol{V}}_{tropism}, \tag{9}$$

where $\overrightarrow{\boldsymbol{V}}_{default}$ is the heading direction of bud $\mathcal{A}$, $\overrightarrow{\boldsymbol{V}}_{tropism} = (0, -1, 0)$ is the tropism vector, and the optimal direction is $\overrightarrow{\boldsymbol{V}}_{opt} = (1/\sum_{j=1}^{n} w_j) \sum_{i=1}^{n} w_i \overrightarrow{\boldsymbol{V}}_i$. The typical value of α and β is set 1.6 and 0.3. Fig. 6 demonstrates that the weight function can cause the growth toward the branch part easier and smoother.

5.3 Geometry construction and foliage synthesis

So far we have created a 3D skeletal graph that represents the main branching structure of the plant. Then this structure is converted into plant geometry by meshing branch models and attaching leaves to tertiary branches and twigs.

Allometric rule application and skeleton refinement. We reconstruct the polygon model of plant branches by using a set of generalized cylinders. To estimate the radius of the cylinder for each node, we apply the allometric rules described by Equation 5 in a root-to-tip manner to extrapolate the skeleton thickness. First, the root node at the plant base should be identified. Similar to [9], we project the point cloud onto a 2D ground plane ($y = 0$) along the vertical axis, and measure the density of the projected points. Then we form a cluster of the points with higher density, and select the cluster centroid as the root point. Next, we compute a cross-sectional plane that is perpendicular to the stem axis at this skeleton root point. Finally, the nearby points in the original point cloud are projected onto this plane. We utilize the B-spline curve fitting [65] to precisely measure the diameter of the trunk at the root node d_{root}.

Since the extracted point cloud from images is usually noisy and contains outliers, the proposed approach does not guarantee a smooth transition between skeleton nodes. Hence, we apply allometric rules and refine the plant structure either on-the-fly after each growth iteration or after the entire growing process. Two operations can be performed: (1) branch collapse. We traverse each node in the skeletal graph; if the degree of this node is 2 and its adjacent edges share similar orientations and radii, then we collapse this node to its parent node; (2) branch smooth. For a node v with parent node u and child w, if the angle between $\overrightarrow{uv}$ and $\overrightarrow{vw}$ is larger than a threshold ϑ (we set $\vartheta = 20°$), then we subdivide the sharp section between these two edges using a Hermite curve [8].

Leaves generation. To complete the plant model, we attach leaves (quads with alpha textures) or flowers (3D mesh) on the created branching structures according to specified plant species. The leaves are generated at the end of the growth cycle, and distributed at the tip nodes along branch directions. We use the leaf configuration method [40] by considering leaf factors, e.g., the number of leaves at one node, the average leaf length and width, the angle between a leaf and its stem, and the normal direction, etc. In the end, we add textures obtained from photographs of natural leaves to our geometry to enhance visual appearance.

TABLE 2
Data statistics and typical parameter values. $|I|$ is the number of photographs used for reconstruction, $|P|$ is the number of points in the downsampled point cloud. H is plant height. $|B|$ is the number of resulting branches, and l_B is the average branch length. The other parameters are explained in Table 1. Note that the angles ϕ and ψ are given in degrees.

Figures	Plant Species	$\|I\|$	$\|P\|$	$H(m)$	l	ϕ	ψ	γ	$\|B\|$	$l_B(m)$
Fig. 2	Pentagon Maple	24	240K	3.92	0.050	120	38	0.45	1733	0.501
Fig. 7(a)	Peach	36	312K	4.76	0.038	97	37	0.50	693	0.429
Fig. 7(b)	Laurel	18	41K	0.97	0.051	95	35	0.43	1980	0.089
Fig. 7(c)	Maple	42	333K	5.21	0.049	130	34	0.42	1768	0.306
Fig. 8	Prunus Triloba	47	261K	3.12	0.052	87	41	0.43	2352	0.394
Fig. 9	Cercis	21	327K	4.45	0.051	100	38	0.41	2788	0.205

6 Experimental Results

In this section, we first present various experimental results to verify the effectiveness and validity of the proposed framework. Next, we evaluate our algorithm and compare our method with other related state-of-the-art approaches. Our algorithm is implemented in C++ with Qt and OpenGL. All the results presented in this paper are obtained on a PC with Intel i7-3770, 3.40 GHz CPU, 16 GB memory, and a 64-bit Windows 7 operating system. The GPU used for rendering is Nvidia GeForce GTX 760.

6.1 Modeling results

Modeling capability. To demonstrate the modeling capability of our approach, we reconstruct a variety of plants with different types of branching structures and foliage densities, ranging from outdoor trees to indoor small-sized plants. We use a hand-held digital camera (Canon EOS 450D) to capture all the photographs used in this paper, while each set of images contains partially overlapping views of the plants. Figs 2 and 7 illustrate several selected modeling results by showing them side by side with the 2D photographs. Note that for efficient implementation, we downsample the reconstructed point clouds and scale all of them into a boundingbox with a size $10m \times 10m \times 10m$. The data complexity and used parameters for each plant are listed in Table 2. The figures present that the rendered images are similar to the photos. Thus, the plant models are realistic to their 3D shapes. These results demonstrate that our system can generate plant models that are close to real ones.

Fig. 8 shows a modeling result of a tree of Prunus Triloba with heavy occlusion. In this example, the foliage density is high and the interior branches of the upper crown are invisible; thus, the interior point cloud cannot be obtained. Therefore, it needs to model non-visible branches that are important in determining the plant crown shape. Unlike the approach of [17]

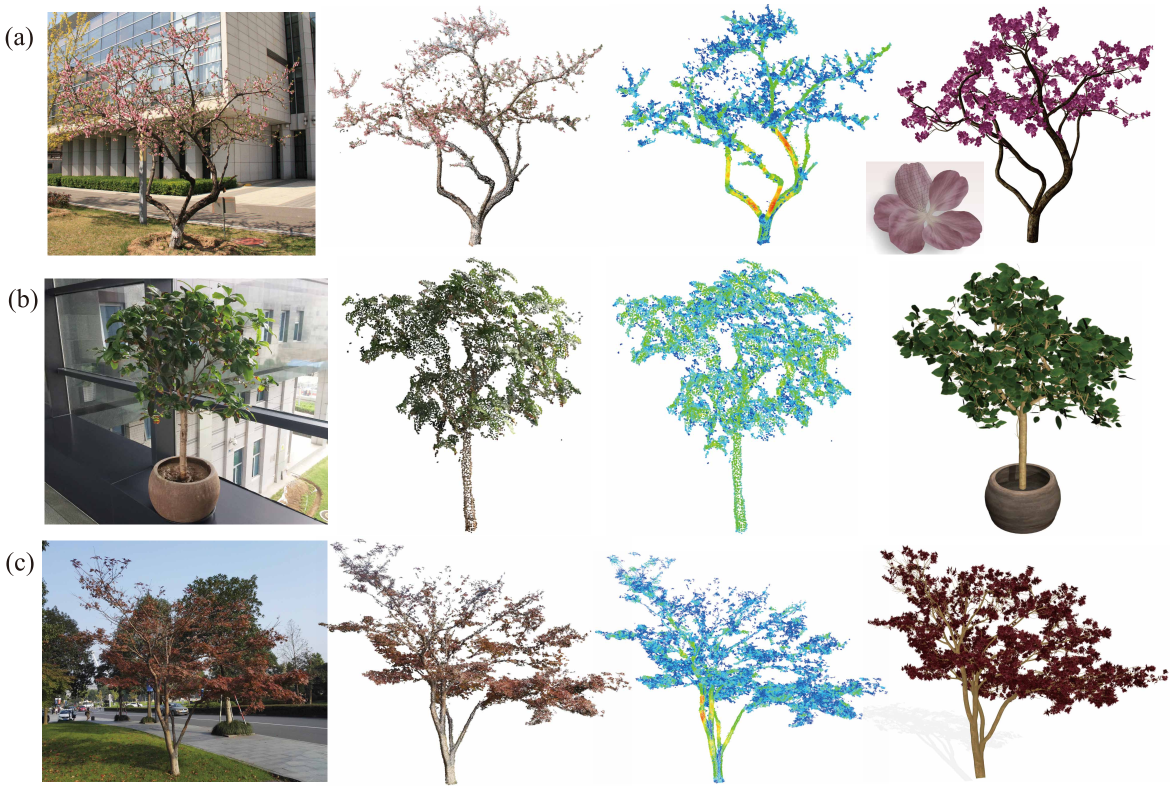


Fig. 7. Modeling results of our method for the plants of: (a) Peach, (b) Laurel, (c) Maple. Left-to-right are one of input images, point cloud, point weight with color coding, textured plant model with foliage.

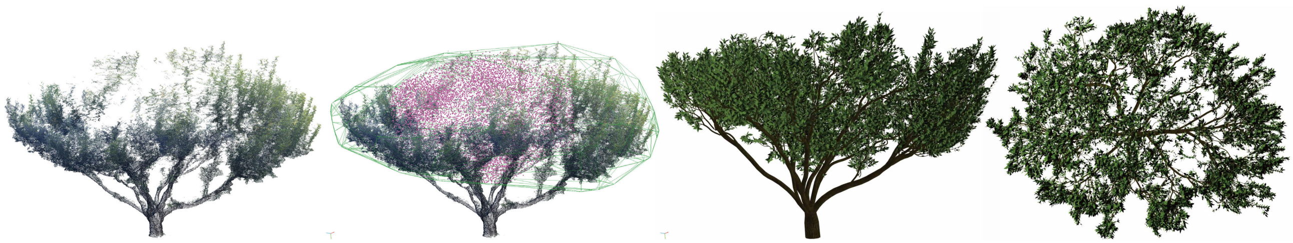


Fig. 8. Modeling a Prunus Triloba tree with self-occlusion. From left to right: incomplete point cloud; convex hull (in line mode) of the crown and the sampled points (shown in pink); reconstructed 3D tree model; top view of the tree model.

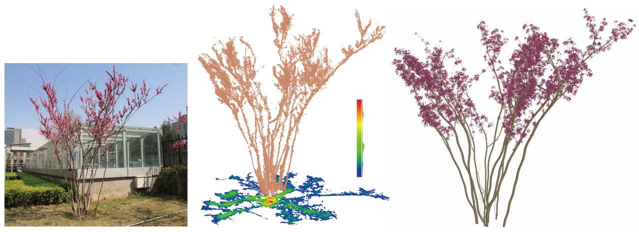


Fig. 9. Modeling of a Cercis plant. From left to right: 2D photograph, reconstructed point cloud and color-coded density of projected points, full plant with foliage.

that builds a library of replication blocks from visible branches, we could apply a simpler and more natural method because our rule-based growth mechanism essentially contains a self-similar property. First, we compute the convex hull of the crown using the reconstructed 3D points. Then we perform Poisson-disk sampling [66] to generate randomly and uniformly distributed points in the interior of the convex hull. Guided by these overdispersed points, our rule-based growth creates branches to fill the crown space, as shown in Fig. 8 (d), which illustrates that the plant representation through our technique has an advantage of including side data and top and internal data.

Non-tree plants modeling. Much of existing works on plant modeling have focused on modeling trees with main trunk and branches obviously observed from outside. However, in addition to trees, many ground-

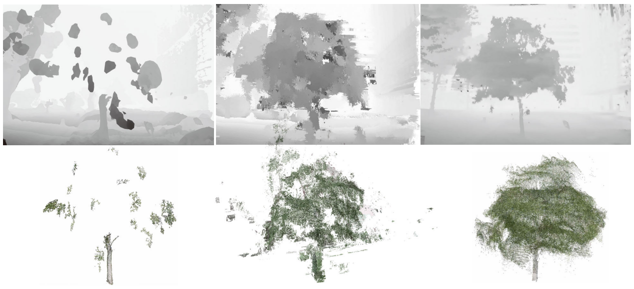


Fig. 10. Comparison of our point cloud reconstruction (right) with a typical MVS method used in image-based tree modeling [17] (left), and a traditional PatchMatch stereo [59] (middle).

covering plants such as bushes and shrubs, reside under trees in large-scale real scenes. The modeling of such ground-covering plants is challenging and has gained little attention; methods like Livny et. al [9] cannot automatically reconstruct these plants.

A modeling example of a Cercis plant with multiple roots is illustrated in Fig. 9. In this case, we compute k roots by selecting the top k cluster centroids according to their density (see the middle of Fig. 9). Then, starting from the multiple roots (each initialized with a separate seed node and with the same parameters), our method can simulate the competition for point clouds between these roots.

6.2 Comparisons

Depth map comparison. As we have discussed in the related work section, the SFM and PMVS approaches, which have been used in plant reconstruction [11], [17], suffer from the accurate matching problem for plant images, and cannot produce adequate points to satisfy our plant modeling requirements. In the first column of Fig. 10, only a small amount of the trunk and branch points are matched and reconstructed.

In this paper, we focus on depth map based plant point cloud reconstruction methods, in which pixels along epipolar lines are used to generate depth maps instead of merely matching key point features. We compute the dense depth map for each plant image by improving the PatchMatch stereo [59], whose slanted support windows can correctly reconstruct these slanted surfaces with sub-pixel precision. In this way, our branches can be generated automatically by growing towards the point clouds. Therefore, our reconstructed tree model can be coincident with ground truth in different perspectives. We also provide a comparison with [59] in Fig. 10, and it shows that our approach is more robust to generate point clouds for multi-view plant images.

Qualitative modeling comparison. Next, we compare our algorithm with the global tree reconstruction method proposed by Livny et. al [9], which robustly reconstructs skeletal structures of trees from point clouds. Given the same input point cloud (see Fig. 11), we apply our algorithm and the method of [9] to reconstruct 3D tree models. The data statistics is listed in Table. 3. This example demonstrates that, compared to [9], our method can reconstruct visually similar 3D tree models. The detailed comparison shows that the branches of our result can better match the real point data (see the branches in green rectangles), although the straightness of our branches is not as good as [9].

TABLE 3
Statistics for comparison. $|P|$ is the number of points. H is plant height. $|B|$ is the number of resulting branches, and l_B is the average branch length. D is the average mesh-to-scan distance, and σ is the standard deviation of this distance.

Figures	Methods	$\|P\|$	$H(m)$	$\|B\|$	$l_B(m)$	$D(cm)$	$\sigma(\%)$
Fig. 11	Livny's [9]	38K	20.13	3921	1.052	0.36	4.3
Fig. 11	Ours	38K	20.13	2307	1.382	0.28	4.7
Fig. 14	Livny's [9]	368K	11.58	2691	0.543	0.55	6.2
Fig. 14	Ours	368K	11.58	1170	0.598	0.17	5.1

Further, previous reconstruction approaches [9], [10], [17] only obtained a single static 3D tree model. By contrast, our method shows the advantages of both data-driven and rule-based approaches. Thus, we can not only accomplish a reconstruction mission, but also model the growing process. In Fig. 12, we show the modeling process of a tree. In this process, we apply allometric rules and refine the tree structure on-the-fly; hence, some unreasonable branches can be shed once they do not satisfy the allometric rules. Thus, our method has a great potential for solving the problem of exact inverse plant modeling from existing real-world data. Another advantage of our approach over other methods, which reconstruct a plant graph from a point cloud or images, is that we can also generate similar plants by slightly adjusting the parameters. This advantage is from the procedural model we used in our approach. Such an example is shown in Fig. 13.

Quantitative comparison. In order to compare the quantitative performance of our method with that of the state-of-the-art [9], we conduct two evaluation types, namely, model-based and scan-based evaluation. First, we collect 10 synthetic plant models from online resources, such as Xfrog library [1], which are used as ground truth. For each model, we render 20 images from uniformly distributed views. We apply our method to extract the point cloud and reconstruct each model. Finally, the similarity S_T of each reconstructed plant and the ground truth model is computed. We use the similarity distance proposed by Stava et al. [16], which incorporates shape, geometry, and structure distance. This measurement evaluates the visual and structural differences between the two models, where a value $S_T = 1$ reflects the exact similarity and $S_T = 0$ means no similarity at all. Our experiments show that the average model similarity

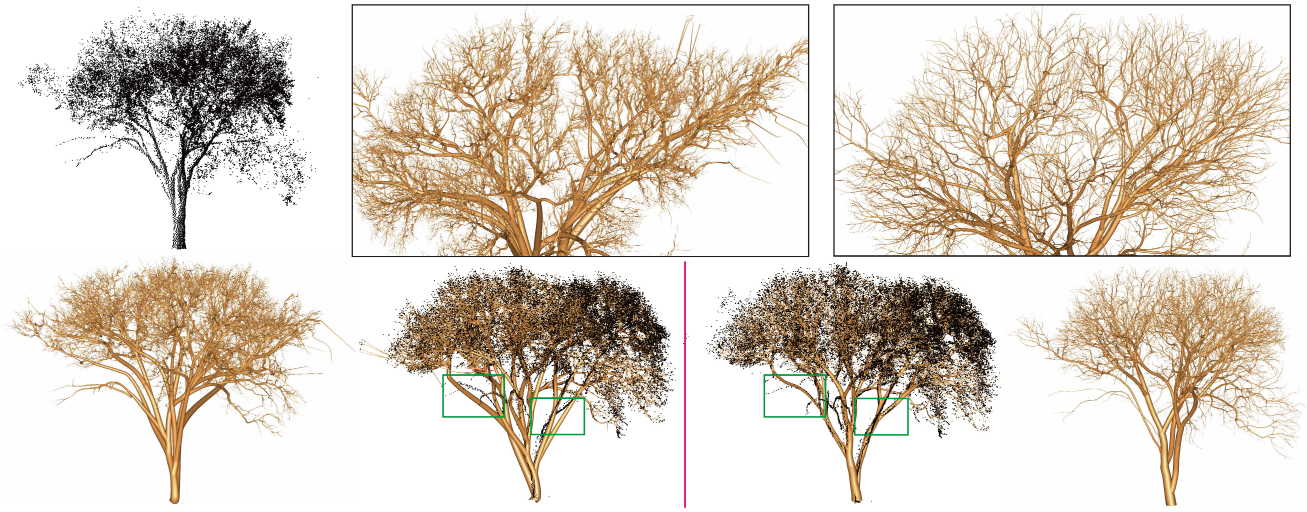


Fig. 11. Comparison of plant reconstruction with a global approach [9]. Top left shows the input point cloud, and top right are the local results (crown part) of [9] and our method. Bottom row: left two images are the reconstruction results of [9] shown in different views, right two images are our results.

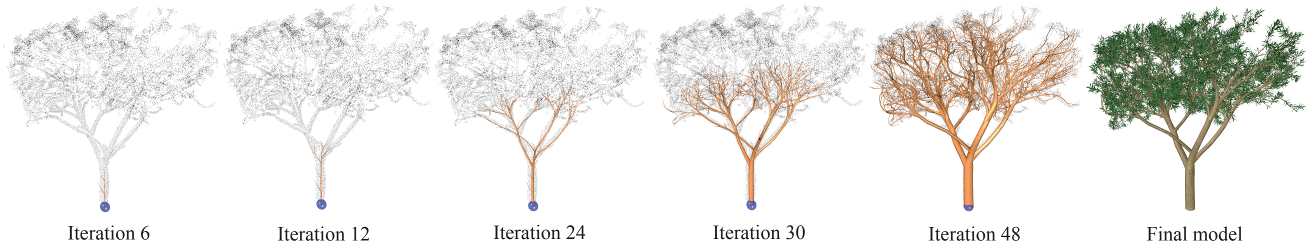


Fig. 12. Modeling of a tree guided by point cloud through simulating the process of an expanding tree structure. We show the resultant trees at different iteration steps, where the root point is represented using a purple sphere.

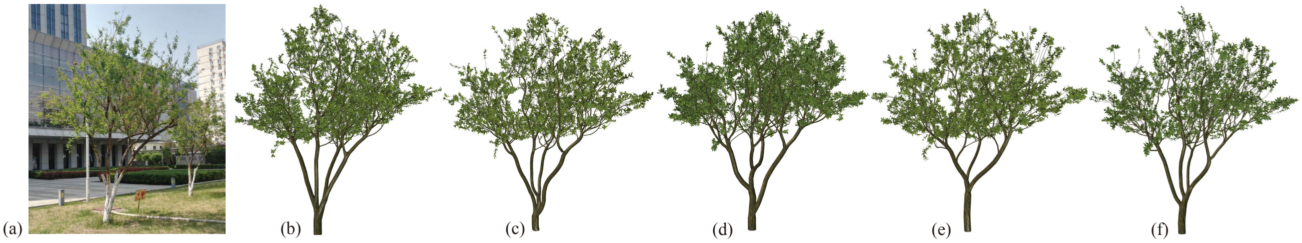


Fig. 13. (a): reference photo, (b): our default reconstruction result, (c-f): similar plants generated by slightly adjusting the parameters while being constrained to the shape of the real data.

of our method is 0.893, whereas the average similarity of [9] is 0.815. The higher similarity indicates our method is more effective to maintain the botanical fidelity of plants.

Next, we scan several trees without leaves in the winter using a Leica ScanStation P20 laser scanner, where the scanned high-precision point cloud can be used as ground truth. We reconstruct the models using our algorithm and the method of [9]. For each vertex in the model, we compute the distance between the vertex and its closest point in the point cloud. The mesh-to-scan distance map of one tree example (its height is 11.58 m) is depicted in Fig. 14. The average distance and standard deviation of Livny et. al [9] are 0.55 cm and 6.2%, whereas ours are 0.17 cm and 5.1%. These visual and quantitative analysis results validate that our method has a better performance.

6.3 Limitations

We successfully used the proposed method to generate various realistic plant models from images. Nevertheless, our method also has limitations. First, although the parameters for different plant species can be pre-generated or specified in a database, we have to determine to which species the plant to be reconstructed belongs to. However, many studies [46],

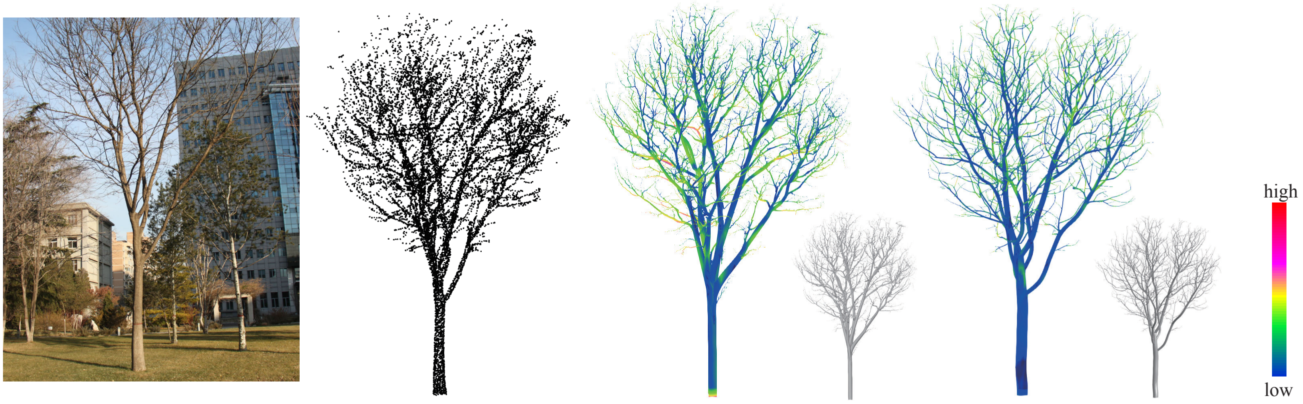


Fig. 14. Quantitative evaluation using a mesh-to-scan distance map. From left to right: 2D photograph, ground truth point cloud, distance map of the approach [9] and distance map of our method.

[67] have focused on the problem of plant species classification, which may be helpful for us. In addition, we may still need to slightly adjust the parameters for a specific plant because different instances of the same plant species often have different geometric shapes due to the random factors that affect the growth process in the nature. One possible solution may be that we compute the parameters of each instance from real data (images or point cloud), inspired from the inverse procedural modeling approach [16].

Second, our method works efficiently for a large amount of branching plants, but fails to model other vegetations that are not captured by our parametric plant representation, such as flowers and climbing plants (e.g., vines, ivy) [68], [69]. For these plants, additional botanical rules and growth parameters would be necessary to consider.

7 Conclusion and Future Work

We presented a novel framework for reconstructing realistic 3D plant models from images. The key feature of our method is the combination of both procedural modeling and data-driven reconstruction approaches. This combination allows generating visually promising plants with precise skeletal structures, while maintaining the botanical fidelity of the plants. Specifically, a dense point cloud is first obtained from multi-view images by exploiting a binocular stereovision approach. A procedural plant modeling guided by real point cloud is then performed. Our technique has an advantage of modeling an integral plant model from multiple side view images, in which a possible data losing from top view can be remedied by integral plant models. This advantage denotes a valuable application in computer graphics-related requirements. The experimental results validate our modeling approach for various plants. The comparison with the state-of-the-art methods shows the advantage of our method for simulating the plant modeling process.

Several possible open problems exist for future work. First, we would like to generalize our method to model larger scale scenes, such as a real ecosystem or even a forest. In this case, we expect to use real data in a deeper way, such as automatic classification and recognition of plant species from images or point clouds. Second, since our modeling approach is rule-based, we are also interested in simulating the dynamic plant behaviors after the environmental factors have changed, which is important for ecology and urban planning applications. Finally, although it is still possible to show the plant growth process with our recovered model, we cannot verify this simulation because we do not have real plant growth history data. We would like to extend our work to plant growth simulation in the future based on the historic data of the growth. Therefore, we should have a link of all historic data in the time axis.

Acknowledgments

We thank anonymous reviewers for their valuable comments. This work is partially funded by the National Natural Science Foundation of China (61331018, 61761003, 61802406, 61772523), the Beijing Natural Science Foundation (4184102), and the CAS Grant (GJHZ1862). Dong-Ming Yan and Zhanglin Cheng are the corresponding authors.

References

[1] O. Deussen and B. Lintermann, *Digital design of nature: computer generated plants and organics*. Springer Science & Business Media, 2006.

[2] X. Liang, V. Kankare, J. Hyyppä, Y. Wang, A. Kukko, H. Haggrn, X. Yu, H. Kaartinen, A. Jaakkola, F. Guan, M. Holopainen, and M. Vastaranta, "Terrestrial laser scanning in forest inventories," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 63–77, 2016.

[3] P. Prusinkiewicz, "Modeling plant growth and development," *Current opinion in plant biology*, vol. 7, no. 1, pp. 79–83, 2004.

[4] A. Lindenmayer, "Mathematical models for cellular interactions in development. ii. simple and branching filaments with two-sided inputs." *Journal of Theoretical Biology*, vol. 18, pp. 300–315, 1968.

[5] H. Honda, "Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body," *Journal of Theoretical Biology*, vol. 31, no. 2, pp. 331 – 338, 1971.

[6] A. Runions, B. Lane, and P. Prusinkiewicz, "Modeling trees with a space colonization algorithm," in *Proceedings of the Third Eurographics Conference on Natural Phenomena*, 2007, pp. 63–70.

[7] P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane, "The use of positional information in the modeling of plants," in *the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01, 2001, pp. 289–300.

[8] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Trans. on Graphics*, vol. 26, no. 4, 2007.

[9] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 29, no. 6, p. 151, 2010.

[10] Z. Wang, L. Zhang, T. Fang, P. T. Mathiopoulos, H. Qu, D. Chen, and Y. Wang, "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, 2014.

[11] L. Lou, Y. Liu, J. Han, and J. H. Doonan, "Accurate multi-view stereo 3d reconstruction for cost-effective plant phenotyping," in *Image Analysis and Recognition: 11th International Conference, ICIAR 2014, Proceedings, Part II*, 2014, pp. 349–356.

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

[13] S. Pirk, O. Stava, J. Kratt, M. A. M. Said, B. Neubert, R. Měch, B. Benes, and O. Deussen, "Plastic trees: Interactive self-adapting botanical tree models," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 50:1–50:10, 2012.

[14] W. Palubicki, K. Horel, S. Longay, A. Runions, R. Mech, and P. Prusinkiewicz, "Self-organizing tree models for image synthesis," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 28, 2009.

[15] R. Wang, Y. Yang, H. Zhang, and H. Bao, "Variational tree synthesis," *Computer Graphics Forum*, vol. 33, no. 8, pp. 82–94, 2014.

[16] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Mech, O. Deussen, and B. Benes, "Inverse procedural modelling of trees," *Computer Graphics Forum*, vol. 33, no. 6, pp. 118–131, 2014.

[17] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan, "Image-based tree modeling," in *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 26, no. 3, 2007, p. 87.

[18] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. New York, USA: Springer-Verlag New York, Inc., 1990.

[19] P. Prusinkiewicz, M. James, and R. Měch, "Synthetic topiary," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '94, 1994, pp. 351–358.

[20] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Mech, "L-systems: from the theory to visual models of plants," in *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, vol. 3, 1996, pp. 1–32.

[21] J. Bloomenthal, "Modeling the mighty maple," *SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 305–311, 1985.

[22] P. E. Oppenheimer, "Real time design and animation of fractal plants and trees," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 55–64, 1986.

[23] J. Weber and J. Penn, "Creation and rendering of realistic trees," ser. SIGGRAPH '95. ACM, 1995, pp. 119–128.

[24] P. de Reffye, C. Edelin, J. Françon, M. Jaegerl, and C. Puech, "Plant models faithful to botanical structure and development," *ACM SIGGRAPH*, vol. 22, pp. 151–158, 1988.

[25] B. Beneš, N. Andrysco, and O. Štava, "Interactive modeling of virtual ecosystems," in *Proceedings of the Fifth Eurographics conference on Natural Phenomena*, 2009, pp. 9–16.

[26] J. Kim and H. Cho, "Efficient modeling of numerous trees by introducing growth volume for real-time virtual ecosystems," *Computer Animation and Virtual Worlds*, vol. 23, no. 3-4, pp. 155–165, 2012.

[27] B. Beneš, O. Št'ava, R. Měch, and G. Miller, "Guided procedural modeling," in *Computer Graphics Forum*, vol. 30, no. 2. Wiley Online Library, 2011, pp. 325–334.

[28] S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz, "Treesketch: Interactive procedural modeling of trees on a tablet," in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, ser. SBIM '12, 2012, pp. 107–120.

[29] L. Xu and D. Mould, "A procedural method for irregular tree models," *Computers & Graphics*, vol. 36, no. 8, pp. 1036 – 1047, 2012.

[30] L. Xu and D. Mould, "Procedural tree modeling with guiding vectors," *Computer Graphics Forum*, vol. 34, no. 7, pp. 47 – 56, 2015.

[31] C. Li, O. Deussen, Y.-Z. Song, P. Willis, and P. Hall, "Modeling and generating moving trees from video," *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, vol. 30, no. 6, pp. 127:1–127:12, 2011.

[32] 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 2001.

[33] A. Reche-Martinez, I. Martin, and G. Drettakis, "Volumetric reconstruction and interactive rendering of trees from photographs," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 23, no. 3, pp. 720–727, 2004.

[34] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang, "Image-based plant modeling," in *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 25, no. 3, 2006, pp. 599–604.

[35] P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan, "Single image tree modeling," *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, vol. 27, no. 5, pp. 108:1–108:7, 2008.

[36] B. Neubert, T. Franken, and O. Deussen, "Approximate image-based tree-modeling using particle flows," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 26, no. 3, 2007.

[37] D. Bradley, D. Nowrouzezahrai, and P. Beardsley, "Image-based reconstruction and synthesis of dense foliage," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 32, no. 4, p. 74, 2013.

[38] J. Guénard, G. Morin, F. Boudon, and V. Charvillat, "Reconstructing Plants in 3D from a Single Image Using Analysis-by-Synthesis," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, 2013, vol. 8033, pp. 322–332.

[39] N. Pfeifer, B. Gorte, D. Winterhalder, *et al.*, "Automatic reconstruction of single trees from terrestrial laser scanner data," in *Proceedings of 20th ISPRS Congress*, 2004, pp. 114–119.

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

[41] F. Aiteanu and R. Klein, "Hybrid tree reconstruction from inhomogeneous point clouds," *The Visual Computer*, vol. 30, no. 6, pp. 763–771, 2014.

[42] A. Bucksch and R. Lindenbergh, "CAMPINO-A skeletonization method for point cloud processing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 63, no. 1, pp. 115 – 127, 2008.

[43] A. Bucksch, R. C. Lindenbergh, and M. Menenti, "Skeltre-fast skeletonisation for imperfect point cloud data of botanic trees," in *Proceedings of the 2nd Eurographics Conference on 3D Object Retrieval*, 2009, pp. 13–20.

[44] J.-F. Côté, J.-L. Widlowski, R. A. Fournier, and M. M. Verstraete, "The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar," *Remote Sensing of Environment*, vol. 113, no. 5, pp. 1067 – 1081, 2009.

[45] 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. CAD/Graph. Comput.*, 2009, pp. 572–575.

[46] Y. Livny, S. Pirk, Z. Cheng, F. Yan, O. Deussen, D. Cohen-Or, and B. Chen, "Texture-lobes for tree modeling," in *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2011, p. 1.

[47] P. Raumonen, M. Kaasalainen, M. ?kerblom, S. Kaasalainen, H. Kaartinen, M. Vastaranta, M. Holopainen, M. Disney, and P. Lewis, "Fast automatic precision tree models from terrestrial laser scanner data," *Remote Sensing*, vol. 5, no. 2, p. 491, 2013.

[48] S. Friedman and I. Stamos, "Automatic procedural modeling of tree structures in point clouds using wavelets," in *International Conference on 3D Vision*. IEEE, 2013, pp. 215–222.

[49] Y. Li, X. Fan, N. J. Mitra, D. Chamovitz, D. Cohen-Or, and B. Chen, "Analyzing growing plants from 4d point cloud

data,” *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, vol. 32, no. 6, pp. 157:1–157:10, 2013.

[50] Z. Wang, L. Zhang, T. Fang, X. Tong, P. T. Mathiopoulos, L. Zhang, and J. Mei, “A local structure and direction-aware optimization approach for three-dimensional tree modeling,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4749–4757, 2016.

[51] G. Wang, H. Laga, N. Xie, J. Jia, and H. Tabia, “The shape space of 3d botanical tree models,” *ACM Trans. on Graphics*, vol. 37, no. 1, p. 7, 2018.

[52] H. C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, vol. 293, no. 5828, p. 133, 1981.

[53] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: Exploring photo collections in 3d,” *ACM Trans. on Graphics*, vol. 25, no. 3, pp. 835–846, July 2006.

[54] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.

[55] S. Shen, “Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes,” *IEEE Trans. on Image Processing*, vol. 22, no. 5, pp. 1901–1914, 2013.

[56] C. R. Michael Bleyer and C. Rother, “Patchmatch stereo - stereo matching with slanted support windows,” in *Proceedings of the British Machine Vision Conference*, 2011, pp. 14.1–14.11.

[57] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[58] N. Snavely, S. M. Seitz, and R. Szeliski, “Modeling the world from internet photo collections,” *Int. Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[59] 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. on Image Processing*, vol. 24, no. 7, pp. 2182–2196, July 2015.

[60] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, “Consistent depth maps recovery from a video sequence,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, June 2009.

[61] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, June 2013.

[62] P. Jaccard, “Eine neue auffassung ueber die ursachen des dickenwachstums der baeume,” *Naturwiss. Z. fuer. Landwirtschaft*, no. 13, pp. 321–360, 1913.

[63] B. Zeide, “Analysis of growth equations,” *Forest Science*, vol. 39, no. 3, pp. 594–616, 1993.

[64] L. Banin, T. Feldpausch, O. Phillips, T. Baker, J. Lloyd, K. Affum-Baffoe, E. Arets, N. Berry, M. Bradford, R. Brienen, *et al.*, “What controls tropical forest architecture? testing environmental, structural and floristic drivers,” *Global Ecology and Biogeography*, vol. 21, no. 12, pp. 1179–1190, 2012.

[65] L. You, S. Tang, X. Song, Y. Lei, H. Zang, M. Lou, and C. Zhuang, “Precise measurement of stem diameter by simulating the path of diameter tape from terrestrial laser scanning data,” *Remote Sensing*, vol. 8, no. 9, p. 717, 2016.

[66] J. Guo, D.-M. Yan, L. Chen, X. Zhang, O. Deussen, and P. Wonka, “Tetrahedral meshing via maximal Poisson-disk sampling,” *Comp. Aided Geom. Design*, vol. 43, pp. 186–199, 2016.

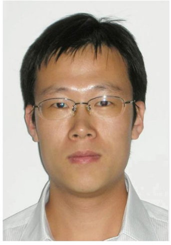
[67] T. Key, T. A. Warner, J. B. McGraw, and M. A. Fajvan, “A comparison of multispectral and multitemporal information in high spatial resolution imagery for classification of individual tree species in a temperate hardwood forest,” *Remote Sensing of Environment*, vol. 75, no. 1, pp. 100–112, 2001.

[68] B. Benes and E. U. Millán, “Virtual climbing plants competing for space,” in *Computer Animation, 2002. Proceedings of.* IEEE, 2002, pp. 33–42.

[69] T. Hädrich, B. Benes, O. Deussen, and S. Pirk, “Interactive modeling and authoring of climbing plants,” in *Computer Graphics Forum*, vol. 36, no. 2, 2017, pp. 49–61.

Jianwei Guo is an assistant researcher in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences(CASIA). He received his Ph.D. degree in computer science from Institute of Automation, Chinese Academy of Sciences in 2016, and bachelor degree from Shandong University in 2011. His research interests include computer graphics, geometry processing, and plant modeling.

Shibiao Xu received the B.S. degrees from Beijing University of Posts and Telecommunications in 2009, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences in 2014. He is currently an associate professor in the Institute of Automation, Chinese Academy of Sciences. His current research interests include computer vision, image based three-dimensional scene reconstruction and understanding.



Dong-Ming Yan is an associate professor in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences(CAS). He received his Ph.D. degree in computer science from Hong Kong University in 2010, and his master and bachelor degrees in computer science and technology from Tsinghua University in 2005 and 2002, respectively. His research interests include computer graphics, geometric processing, and visualization.

Zhanglin Cheng received the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CAS) in 2008. He is currently an Associate Professor with the Visual Computing Research Center (VCC), Shenzhen Institutes of Advanced Technology (SIAT), CAS. His research interests include computer graphics and visualization.

Marc Jaeger is a senior researcher in GreenLab AMAP Laboratory in CIRAD and an associated researcher of Montpellier University LIRMM ICAR team. Graduate from University Louis Pasteur of Strasbourg (PhD, 1988), he specialized on plant growth simulation and visualisation at CIRAD Amap laboratory. His current research topics concern mainly computational plant and landscape and their visualisation.

Xiaopeng Zhang is a Professor in National Laboratory of Pattern Recognition at Institute of Automation, Chinese Academic of Sciences (CAS). He received his Ph.D. degree in Computer Science from Institute of Software, CAS in 1999. He received the National Scientific and Technological Progress Prize (second class) in 2004. His main research interests include computer graphics and image processing.