Realistic Procedural Plant Modeling Guided by 3D Point Cloud

Jianwei Guo
NLPR, Institute of Automation, CAS

Zhanglin Cheng*
Shenzhen VisuCA Key Lab, SIAT, CAS

Shibiao Xu
NLPR, Institute of Automation, CAS

Xiaopeng Zhang
NLPR, Institute of Automation, CAS

Figure 1: Modeling of a small scene from street-level scanned data. The images show a photo of the scene (top left), point cloud (top right) and generated plant models with textured leaves (bottom).

CCS CONCEPTS

• **Computing methodologies → Computer graphics**; *Computational Geometry and Object Modeling*;

KEYWORDS

procedural plant generation, point cloud, rule-based modeling

ACM Reference format:
Jianwei Guo, Zhanglin Cheng, Shibiao Xu, and Xiaopeng Zhang. 2017. Realistic Procedural Plant Modeling Guided by 3D Point Cloud. In *Proceedings of SIGGRAPH '17 Posters, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: 10.1145/3102163.3102193

1 INTRODUCTION

Plants are ubiquitous in the nature, and realistic plant modeling plays an important role in a variety of applications. Over the last decades, an immense amount of efforts have been dedicated to plant modeling. These approaches can be classified into two major categories: procedural modeling [Palubicki et al. 2009; Stava et al. 2014] and data-driven reconstruction approaches (e.g., photographs [Li et al. 2011; Tan et al. 2007] or scanned points [Livny et al. 2010; Xu et al. 2007]). Each approach has its own pros and cons. For example, procedural modeling approaches work well for synthesizing local branch structure details to produce botanically correct trees, but they lack the ability to control the growth of trees under certain shape constraints. While the data-driven approaches might precisely reconstruct skeletal structures, the botanical fidelity of trees are difficult to maintain.

To fully possess advantages of both approaches, in this paper, we present a new modeling framework for generating realistic plants by integrating point cloud analysis with rule-based growth for procedural plant modeling. Using the real point cloud data as soft constraints, we fit a parametric tree model to simulate the plant growing progress. Our method makes several important contributions to the research on plant modeling: (1) enriching the tree generation literature by building connections between virtual tree modeling and data-driven tree reconstruction; (2) generating ground-covering non-tree plants, e.g., bushes and shrubs (see Figure 1), which have gained little attention and cannot be reconstructed by previous approaches.

2 TECHNICAL APPROACH

Our algorithm is based on the observation that obtained point cloud represents plant growing result controlled by external environmental and internal factors. Inspired by the space colonization [Palubicki et al. 2009], we regard the points as a kind of resources, which could

*Corresponding author: zhanglin.cheng@gmail.com

SIGGRAPH '17 Posters, Los Angeles, CA, USA
 978-1-4503-5015-0/17/07…$15.00
DOI: 10.1145/3102163.3102193

guide the growing of the buds. Starting from seeds located at the root, we improve a parametric procedural model [Palubicki et al. 2009] to simulate plant growing.

2.1 Parametric plant representation

We first create a parametric skeletal structure that is powerful enough to generate a variety of plant species. Various procedural modeling methods exist for extracting branch structures, and theoretically, any parameter-driven approaches can be integrated into our framework. However, to avoid tuning too many parameters [Stava et al. 2014], a plant model with 5 parameters (see Table 1) is used for representing the skeletal structure in our work. These parameters are pre-generated and stored in a species library. Please note that the growth rate ρ can be computed automatically in our method by using a logistic growth equation.

Table 1: Botanical parameters used for representing plant skeletal structure.

Paras.	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 rate	number of internodes generated on a single shoot during one growth cycle
γ	diameter coefficient	branch thickness transmission coefficient

2.2 Rule-based modeling guided by point cloud

We now integrate above parametric representation into a rule-based growing mechanism by using L-systems. When implementing this L-systems, [Palubicki et al. 2009] iteratively simulates the space competition between growing branches. They assume that points around each bud are uniformly distributed, and the growing speed is equal at every stage. However, it is not true in real growing process. To adapt this model to the real point cloud, we make two significant improvements.

First, we present an automatic method to adaptively compute the growing distance at each stage. Previous studies by botanists have shown that a plant grows in such a way that the growth increases exponentially when the plant is young and decreases as the plant approaches its asymptotic maximum growth capacity. To simulate this kind of nonlinear growing, we apply the well-known Logistic model which describes the population growth when the environment is constrained by limited resources. Given the internode length l, let H_{max} be the total length from the leaf to the plant root: $H_{max} = K \cdot l$. In a sense, H_{max} approximates the height of the plant. Then in each growth iteration time t, we use the Logistic model to compute the expected tree height H_t at time t. Thus we can derive growth rate ρ from H_t. Finally, the integer part of ρ determines the influence distance $d_i = \lfloor \rho \rfloor \cdot l$ at each stage.

Next, in the work of [Palubicki et al. 2009], they treat every point in perception volume equally. That is to say, each point has the same weight for skeleton computation. However, we find that the points on branches usually have larger influence on the skeleton growth than leaf points. Therefore, in our method, each point is equipped with two attributes: 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 $\mathbf{p}_i$. Then we search in the neighbor of $\mathbf{p}_i$, and count the number of points $\mathbf{p}_j$ that has similar location and eigen vector with $\mathbf{p}_i$. The number of similar neighbors is denoted as weight w_i. The actual growth direction of each bud $\mathcal{A}$ is computed as the follows: $\vec{V}_{\mathcal{A}} = \vec{V}_h + \alpha \vec{V}_{opt} + \beta \vec{V}_t$, where $\vec{V}_h$ is the heading direction, $\vec{V}_t = (0, -1, 0)$ is the tropism vector, and the optimal direction is calculated as the average of normalized vectors $\vec{V}_i$ formed by the bud and neighboring points: $\vec{V}_{opt} = \frac{1}{\sum_{j=1}^{n} w_j} \sum_{i=1}^{n} w_i \vec{V}_i$.

Figure 2: We show the color-coded weight image (warmer color indicates higher weight), as well as modeling results without/with the weight.

So far we have created a 3D skeletal graph that represents the main branching structure of the plant. Finally, this structure is converted into plant geometry by meshing branch models based on allometric rules and by attaching leaves to tertiary branches and twigs according to specified plant species.

3 CONCLUSION AND FUTURE WORK

We present a rule-based framework for generating naturally-looking plant models from real point cloud. In the future, we would like to address automatic classification and modeling of larger scale scenes, such as a real ecosystem.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (Nos. 61331018, 61379091, 61372168 and 61571439).

REFERENCES

Chuan Li, Oliver Deussen, Yi-Zhe Song, Phil Willis, and Peter Hall. 2011. Modeling and Generating Moving Trees from Video. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 30, 6, Article 127 (2011), 12 pages.

Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. 2010. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 6 (2010), 151.

Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, RadomÍŕr Mech, and Przemyslaw Prusinkiewicz. 2009. Self-organizing tree models for image synthesis. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 28 (2009). Issue 3.

Ondrej Stava, Sören Pirk, Julian Kratt, Baoquan Chen, Radomir Mech, Oliver Deussen, and Bedrich Benes. 2014. Inverse Procedural Modelling of Trees. *Computer Graphics Forum* 33, 6 (2014), 118–131.

Ping Tan, Gang Zeng, Jingdong Wang, Sing Bing Kang, and Long Quan. 2007. Image-based tree modeling. In *ACM Trans. on Graphics (Proc. SIGGRAPH)*, Vol. 26. 87.

Hui Xu, Nathan Gossett, and Baoquan Chen. 2007. Knowledge and Heuristic-based Modeling of Laser-scanned Trees. *ACM Trans. on Graphics* 26, 4, Article 19 (2007).