A Procedural Model for Diverse Tree Species

Rama Karl Hoetzlein
Quanta Sciences
Ithaca, New York, USA
ramahoetzlein@gmail.com

ABSTRACT

The modeling of trees represents a unique and classical challenge in computer graphics. Models of 3D trees must express the form, complexity, structure, growth and diversity of real trees. Presently the most common methods for the modeling of 3D trees include a) user-based creative modeling, b) direct geometric capture such as LIDAR and photogrammetry, or c) indirect methods such as machine learning from images. These techniques often require significant human effort, large amounts of data, considerable computation resources, or any of the above. While there are methods that consider the direct procedural generation of trees, current models often require some human supervision to focus on naturally plausible variants. Instead, our approach is to construct a botanically-inspired, harmonic, procedural model for trees which directly produces realistic yet diverse trees.

KEYWORDS

procedural modeling, trees, vegetation, particle system

1 INTRODUCTION

The modeling of trees and vegetation is important to a wide range of applications from motion pictures, to video games, to terrain mapping and visualization. Understanding the form of trees requires an investigation of both geometry and biology. Early methods, such as L-systems, focused on the hierarchical and fractal geometry of trees yet appear unusually rigid since there are many physical forces resulting in complex forms. While most modern techniques capture primary growth, in which trees grow at endpoints called apical buds, many techniques miss secondary growth in which the girth of the tree expands outward. Additionally, real tree branches contend with a range of forces such as gravity, wind, sunlight, shade and obstructions. The diversity of tree species reflects a wide variety of natural biological solutions to these physical challenges. Modeling of 3D trees must ideally capture as many of these real phenomena as possible. We briefly review the current techniques for tree modeling and then present our fully automated biologically-inspired tree model.

Trees have been well studied in computer graphics. Significant research has gone into rendering due to the geometric complexity of trees where forests present a unique challenge due to their density. Consequently a wide range of primitives have been applied to render dense vegetation. Boudon et al. presents a survey of methods for modeling and rendering of trees [2]. While our work here generates polygonal models with level-of-detail, we are only indirectly concerned with rendering as our efforts focus primarily on the aspects of procedural modeling.

1.1 L-Systems and Model Primitives

The earliest models of tree structures include L-systems originally developed by Aristid Lindenmayer for cellular filaments [13]. Interestingly, these models could capture parallel leader growth in filaments, that is, simultaneous growth at multiple points. Branching is a natural outcome of L-systems. Kawaguchi considered the form of branching to develop a model for trunk and branch geometry in three dimensions [11]. Max introduced the cone-sphere primitive to address the gap at the trunk-branch transition [16]. By introducing parametric surfaces, Bloomenthal was able to represent the saddle-shaped region between trunk and branches [1]. All of these early models treated branches as a rigid linear primitive, which still generate realistic results for many types of trees.

1.2 User-based Modeling

The specification of arbitrary 3D curves requires user interaction thus leading to a host of methods for the interactive modeling of trees. A method for curved branches using Bezier splines was introduced by Holton [7]. Lintermann and Deussen demonstrate a high level language for interactive plant modeling which builds trees from geometric components [14]. Power et al. model trees with a dynamic spring-like model [19]. Alenda Chang reviews these techniques and observes their influence in developing the commercial SpeedTree and XfrogPlants software now widely used in the games and motion pictures [3]. More recently, sketch-based approaches offer a way to simplify the design process [4]. In general artistic modeling, with creative human input, represents the most compelling workflow for aesthetically designed trees.

1.3 Data-driven Tree Modeling

A variety of techniques for tree modeling based on data-driven sources have appeared recently. Image-based methods such as Neubert et al. approximate the 3D structure of trees from one or more source images [17]. Part-based approaches use detailed 3D scans of portions of trees to construct a complete three dimensional tree [23]. A novel application of neural networks allows for the reconstruct of trees from single images [15]. Finally, direct measurement of complete trees may soon be possible with terrestrial LIDAR (laser based range and imaging) or photogrammetry [22]. The latest techniques adequately capture the geometry of real branches yet still struggle with leaf canopies due to dense occlusion. With over 70,000 tree species a particular challenge with all data-driven techniques is the availability of data for various species.

1.4 Procedural Modeling of Trees

Direct procedural models generate trees with little or no prior data. Early work in this direction by de Reffye introduced botanical realism with nodes and internodes [5]. A recent advancement is the

Figure 1: Modeling a Mountain Juniper (sp. Juniperus scopulorum) with highly curved, twisted branches and evergreen leaves.

space colonization algorithm which guides apical buds via self-avoidance of the existing tree or obstacles, resulting in realistic branch structures [21] [18]. Xu et al. use guide vectors to create shortest paths for tree branches a priori [24] [25]. These techniques require acceleration data structures for collision avoidance. Our method embeds botanical branch constraints for trees directly into the model.

2 A HARMONIC PROCEDURAL MODEL FOR TREES

We develop a compact, botanically-inspired, fully autonomous procedural model for trees. While the existing methods described above for modeling are available they typically require significant human creative input, considerable budgets for 3D assets, complex data structures and algorithms, or access to large data sets.

Our goal was to develop a compact tree model with the following features:

- Procedural: generating output 3D models from parameters
- Compact: few parameters with minimal memory footprint
- Fully autonomous: requiring no user input whatsoever
- Botanically-inspired: based on features of real trees
- Individual variation: able to generate different individuals of the same species
- Species diversity: able to generate different species
- Natural growth: showing realistic animation and growth
- Polygonal output: generating 3D mesh assets directly usable in both polygon and raytracing pipelines
- Level-of-detail: generating LODs for real-time applications
- Simplicity: minimal effort to code, with no optimization or goal-driven minimization
- Efficiency: able to grow and generate trees quickly

The purpose of this prototype model is not to generate the most highly realistic trees - as those are achieved with space colonization algorithms – but to generate practical, useful and detailed tree models quickly with little effort. To that end we demonstrate a method for rapid tree generation.

2.1 Botanical Basis

A biologically accurate model of plant growth must begin with known plant parts; internodes for existing growth, nodes where branches develop, and buds where new growth occurs [5]. Such a model captures growth accurately but is insufficient for a tree-specific shape since the direction of branch growth depends on a wide variety of physical factors.

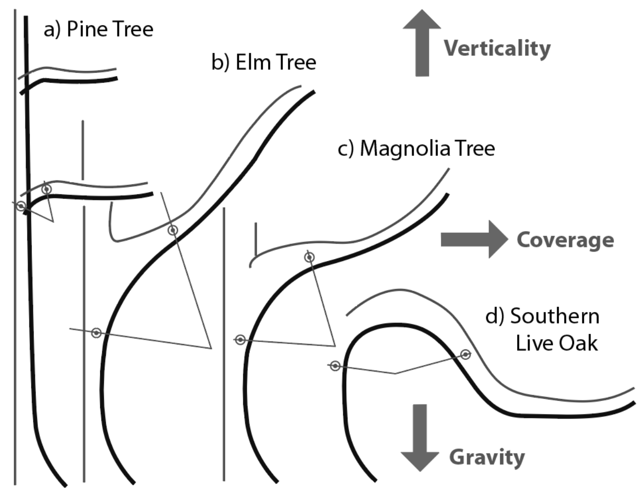

Figure 2: Tree forces motivate the shape of branches. a) Pine trees achieve verticality in the trunk with broad branches for coverage, b) Elm trees achieve both vertically and coverage in the branches, c) Magnolia branches extend outward first then upward, d) Southern Live Oaks curve down due to gravity then slightly upward to avoid the ground. Note that branch shape alone is insufficient to identify a tree since a species may exhibit many behaviors with age.

We consider a global approach by observing that branches of trees follow common patterns based on driving forces (Figure 2). Tree branches are pulled downward by gravity, driven to grow vertically for access to sunlight, and grow laterally for coverage and to compete with neighbors. A mature Southern Live Oak has very broad branches that initially curve downward as the base supports a gravity arch while the middle of the branch curves upward slightly to avoid the ground and the end curves upward to expose the leaves to sunlight. Since the collisions used by space colonization are computationally expensive we instead encode the branch curves using a model that explicitly captures this.

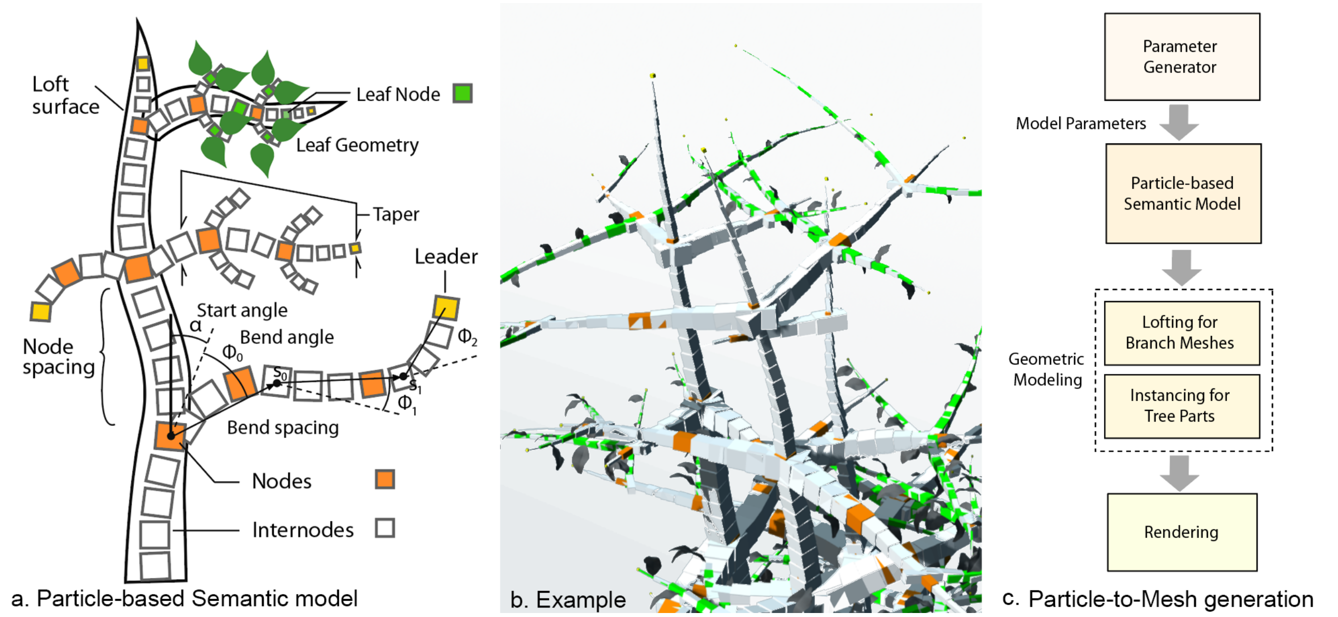

Figure 3: Overview of our procedural model for trees. a) Design of a particle-based semantic model. Apical buds are leader particles (yellow) that grow to produce nodes and internodes. New branches originate at nodes (orange) while branch curves are defined by arc sections that curve upward or downward. b) Illustration of semantics colored on a working example, and c) Overall process for particle-to-mesh generation and rendering.

2.2 Branch Definition

Branches are encoded as a sum of branch features (Figure 4). The dominant term R_i is the branch shape defined as arc sections with different curvatures, see Fig. 3a and 4a. Bend spacing s_k and angle ϕ_k define a parameter pair for each section k, where a positive angle bends the branch upward while negative bends it downward. In these experiments a 4-component vector is sufficient for most force transitions, with vector pairs for each hierarchical level.

To capture angular branches, e.g. Poplar trees, the next term W_i is a linear wandering parameter which adjusts the bud direction by angle w_a at discrete intervals w_s (Fig. 4b). Further realism is achieved with a third harmonic term H_i, e.g. Oak trees, which captures sinusoidal and wavy motion by frequency f_i and amplitude a_i (Fig. 4c).

The complete, time-evolving motion for a branch at level i is defined with quaternions to modify apical bud orientation B:

$$B(t+1) = B(t)\, R_i(t)\, W_i(t)\, H_i(t) \tag{1}$$

where

$$R_i = \phi_k, \text{for } k, s_k < t < s_{k+1} \tag{2}$$

$$W_i = \begin{cases} w_a, & t \mod w_s \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$H_i = sin(t * f_i) * a_i \tag{4}$$

For individual variation within a species a small random variation is added to each parameter. This model explicitly constrains the shape of branch curves to those which bend upward or downward due to physical forces. Arc length spacing is different for each term to provide the greatest flexibility in branch shape.

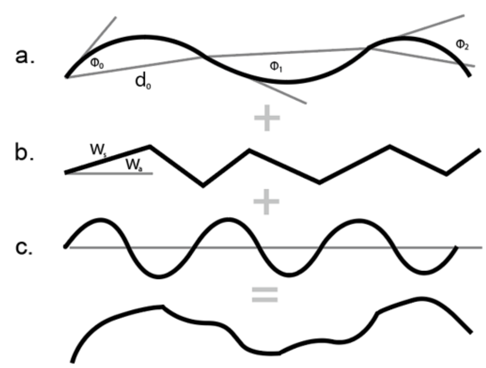

Figure 4: Branch curves are defined as a sum of terms with a) dominant changes in direction as arc segments, b) a linear wandering term for straight features, and c) a harmonic term for sinusoidal features. Each apical leader bud generates a branch by following this trajectory.

Branch generation, that is when new buds occur, is defined by node spacing and probability as separate parameters. A different set is applied to each hierarchy level. The complete model consists of 100 float parameters (400 bytes) composed of 8 general parameters, 36 hierarchical parameters, 24 arc parameters, 8 linear parameters, and 12 harmonic parameters. Model parameters are provided in Appendix A.

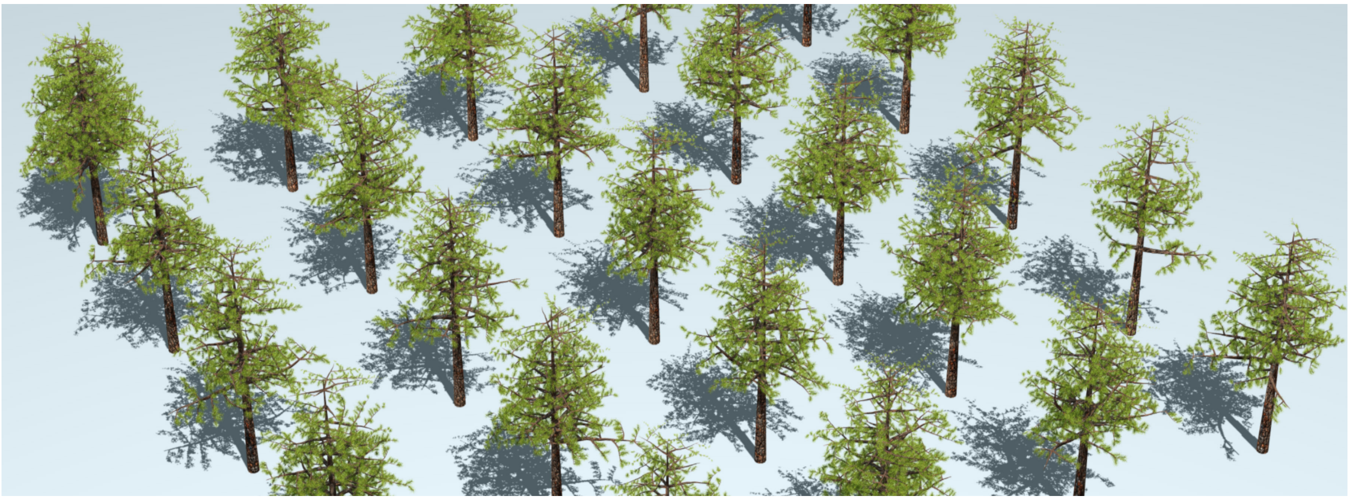

Figure 5: Individual variation is demonstrated with this stand of pine trees. These trees are not instanced, each tree is a unique individual shape for the given species.

2.3 Semantic Particle-based Model

The botanical discussion, and in particular the directional growth of buds, inspires a particle-based model. Particle models are easy to implement and lend themselves well to complex motion [20]. This approach was realized by the author previously to create particle hierarchies [6], yet without naturally-motivated branch shapes or nodes, resulting in imaginative trees similar to Juuso [9].

In this model we assign semantic labels to particles corresponding to plant parts, as shown in Figure 3a (green nodes). Branch nodes are triggered when the apical bud reaches a multiple of the node spacing. New buds are initiated probabilistically. To model secondary growth and natural tapering, internodes begin at a small size and then all internodes in the tree are grown simultaneously on each time step. Full growth is complete when an age limiting parameter is reached.

By assigning additional semantics we can model the many complex parts of a tree. Each additional tree part adds 14 parameters. Leaves, thorns, auxiliary buds, pinecones and flowers can be assigned as semantic labels on particles which are later replaced with generative sub-models or static meshes. The spacing of parts can be set independently to match the spacing of nodes (e.g. aux buds) or other features (e.g. thorns).

2.4 Particles-to-Mesh Geometry

With advances in hardware, polygons are still the de facto standard primitive for rendering pipelines. While particles have been used as primitives for tree rendering one must typically use custom shaders or alternative techniques to render particles directly as tree parts [20] [8]. With the advent of virtualized geometry in 2021, first appearing in Unreal Engine 5, it is now possible to render scenes with billions of triangles while averaging 20 million triangles per frame [10]. This technique is based on real-time mesh simplification and, while well suited to immensely detailed meshes, does not perform well with foliage since a leaf is already an atomic shape of minimal complexity. We believe, however, that techniques such as geometry instancing will be increasingly important, as UE5's Nanite has also introduced micro-polygon rendering. Therefore we focus on reusable meshes as an output primitive.

Rather than render directly from particles we use particles as an underlying representation to generate lofted meshes for branches and geometry instances for leaves. The algorithm identifies contiguous particle-chains from the evolved hierarchy and forms these into lofted filaments. Where a branch occurs a new chain is identified.

A particle-to-mesh pipeline has several benefits. First, bump mapping and displacement can be easily applied to trunk and branches. We demonstrate bump mapping here yet one could also generate displacement dynamically with GPU tessellation. Second, the lofting step provides a simple mechanism for automatic level-of-detail since both the number of branches and their UV resolution is adjustable. Third, all repetitive tree parts (leaves, thorns, pinecones) take advantage of GPU-based geometry instancing for efficient rendering of the foliage. Finally, the output of this process is a geometric mesh with bump and texturing that may be used directly in raster or raytracing pipelines.

3 RESULTS

Several results are demonstrated with this botanically inspired model. First, we show the ability to create both individual and species variations. Second, we automatically generate diverse hypothetical tree species by randomly sampling the parameter space. Third, we demonstrate the ability to model real trees for specific species. Finally, we show the benefits of mesh output by rendering scenes with both rasterization (OpenGL) and raytracing (OptiX) pathways.

3.1 Individual Variation

Our model can distinguish individual variation from species variation. Individual variations within a species have the same average branch angles, lengths, and salient features. For example, the probability of branching at a node may be consistent for a given species

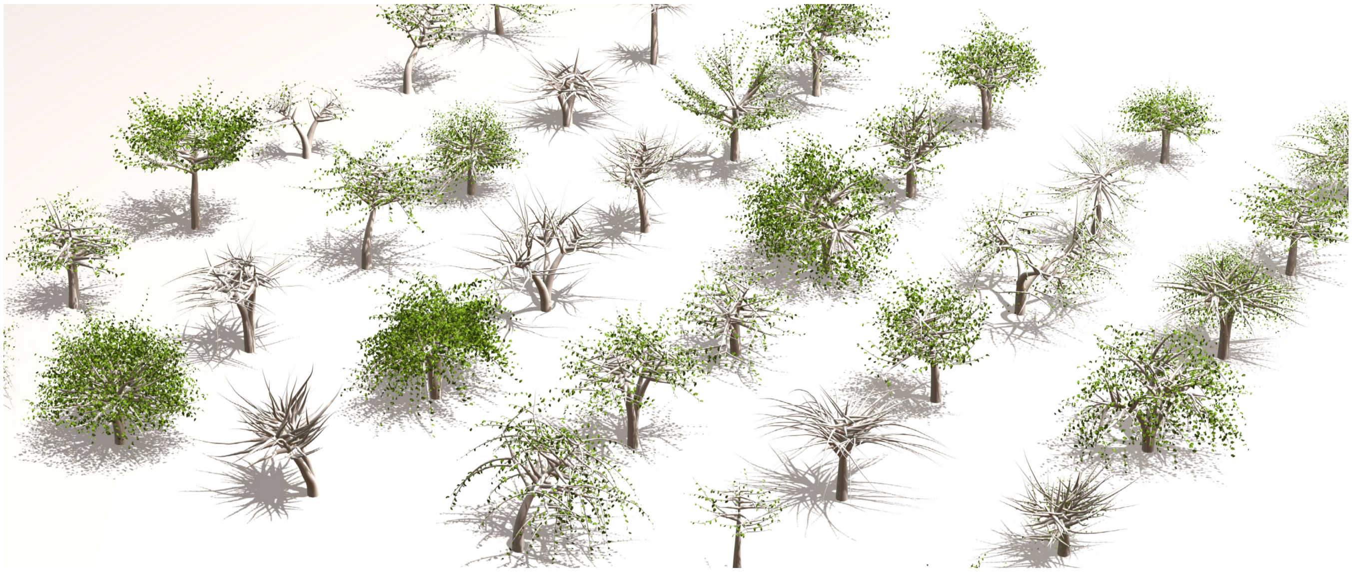

Figure 6: A diverse tree collection created by randomizing the parameter space of our botanically-inspired model. Each example is a plausibly realistic tree since the model embeds natural branch shapes.

yet whether a branch actually occurs defines the individual tree. Minor random variations in the parameters, and deviations in the random walk of leader buds, give rise to individual trees. Figure 5 shows a stand of unique individual Ponderosa pine trees. These are not instanced trees – each one is a unique tree within the species.

3.2 Species Variation

A unique feature of this approach is that directionality changes are explicit in the arc model. Most procedural models for particle-based trees are open ended; requiring some way to limit the feature space through human authorship or real-world data. Here the encoding of branch shape is explicitly tied to the inspired force model. By specifying the minimum and maximum range for each parameter we can randomly sample the parameter space to generate trees autonomously.

Figure 6 shows 36 tree species randomly sampled from the parameter space. Interestingly, nearly every example in the image appears as a plausible natural tree. This presents a distinctly different workflow where one *discovers* trees rather than modeling them. We can imagine a search tool, similar to shape libraries for furniture, in which one is guided through the latent parameter space.

3.3 Modeling of Specific Species

Certain species of trees are more prevalent than others. Pine trees are dominant in many forests. Oak, birch and maple trees are frequently found in temperate meadows. Any system for tree modeling should be able to realistically produce these and more exotic species. Figure 8 shows the rendering of common trees using our model. Parameters were directly tuned for each species, with bark and leaf textures to match.

While a user interface for modeling would be possible this was not our design goal. We focus on a fully parametric model to define branch shape with relatively few parameters suitable for latent space exploration or machine learning.

3.4 Rendering

The particle-to-mesh algorithm produces geometry suited to either rasterization or raytracing. The paper birch tree in Figure 7 was rendered and measured in both OpenGL and OptiX.

Input consists of 100 parameters requiring 400 bytes, plus leaf mesh (8k bytes) and textures (112k). The procedural model generates this tree with 15146 particles in 2.4 seconds. Lofted branches having a circular cross section (with bump mapping) result in 687 unique 3D branches with a loft UV resolution of 16x64 resulting in 1984 tris per mesh and 1,363,008 triangles for all branches.

Parts, such as leaves and flowers, are placed and oriented by our algorithm yet modeled manually. These 3D parts could be procedurally generated as well. The leaves are generated with a low resolution input mesh having 201 triangles, with 2044 geometry instances resulting in 410844 triangles for all leaves.

The total geometry per frame is 1,773,852 triangles. OpenGL renders in 15 msec/frame with an added 11 msec/frame for cascade shadow maps (CSM), for a total time of 26 msec or 38 frames per second. Further improvements are possible with hierarchical culling. The realistic OptiX image in Figure 7 renders with 860 msec/sample, using 8 samples/pixel, in a total of 6.8 seconds.

4 DISCUSSION AND CONCLUSIONS

A procedural model is presented for diverse 3D trees that requires minimal user input, has a small footprint, is efficient to compute, and can generate common and exotic tree species with a botanically inspired model for branch shape.

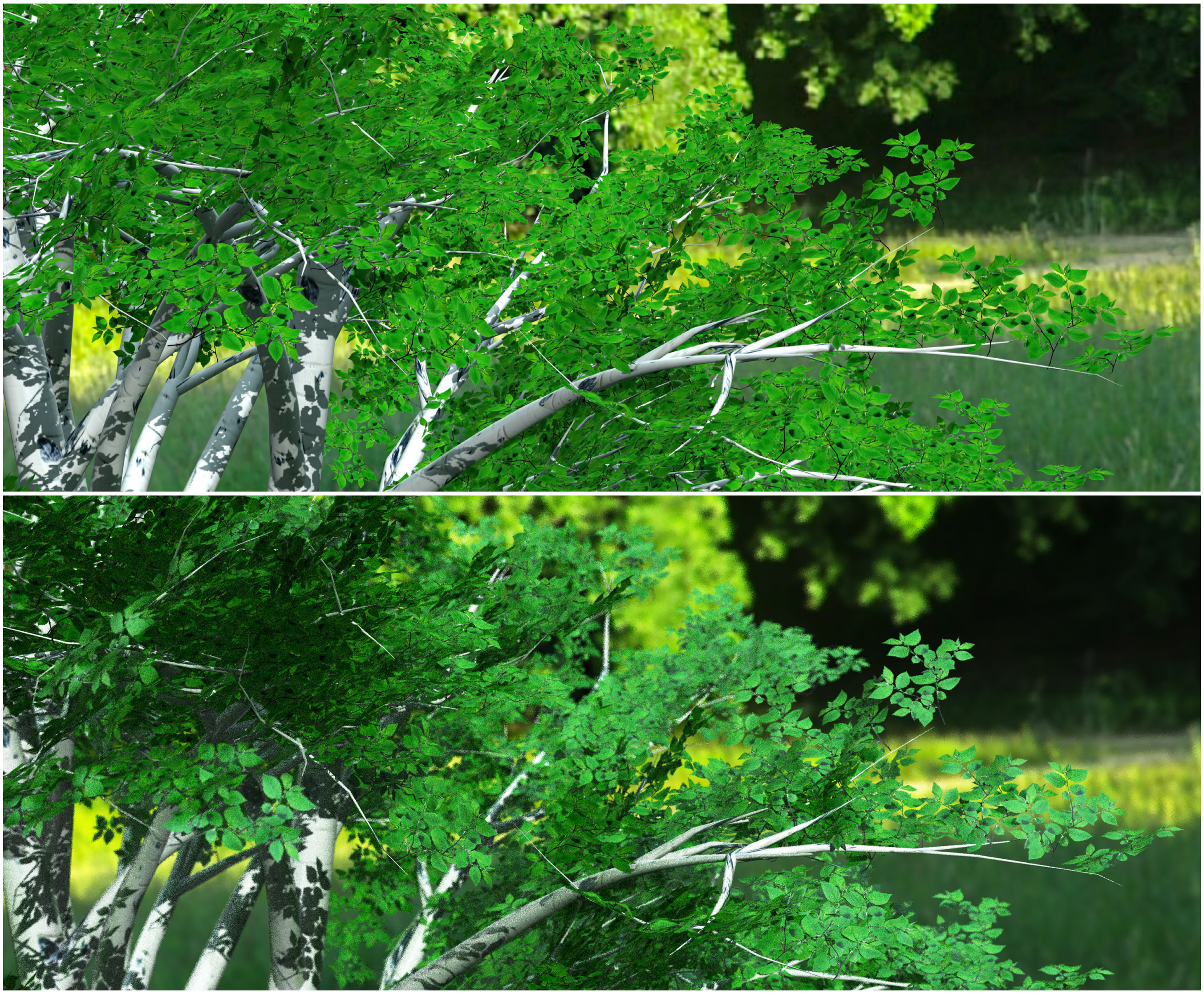

Figure 7: Modeling and rendering of a paper birch tree, *sp. Betula papyrifera*, with the procedural model generating 1.7 million triangles. Rendered using rasterization with OpenGL at 38 fps (top) or by raytracing with Nvidia OptiX in 6.7 seconds (bottom). Our workflow produces meshes easily suited to both.

Limitations include the fact that self-intersection of branches is possible since no space colonization algorithm is present. This could be added to the model if desired. Due to the underlying particle system representation, the addition of a spring model to simulate wind should be straightforward. In the future we hope to model 3D leaves, flowers and other trees parts procedurally as well. Textures and displacement maps might also be procedural with texture synthesis techniques [12]. Machine learning is another valuable area to pursue since a compact, fully parametric model lends itself well to directly learning the parameter space from examples.

Despite advances in tree modeling from captured data we believe that fully procedural models such as this one have an important place in production pipelines for games, motion pictures and visualization as a fast and rapid way to generate diverse content. The exploration of natural tree structures is a complex subject where there is still much to learn.

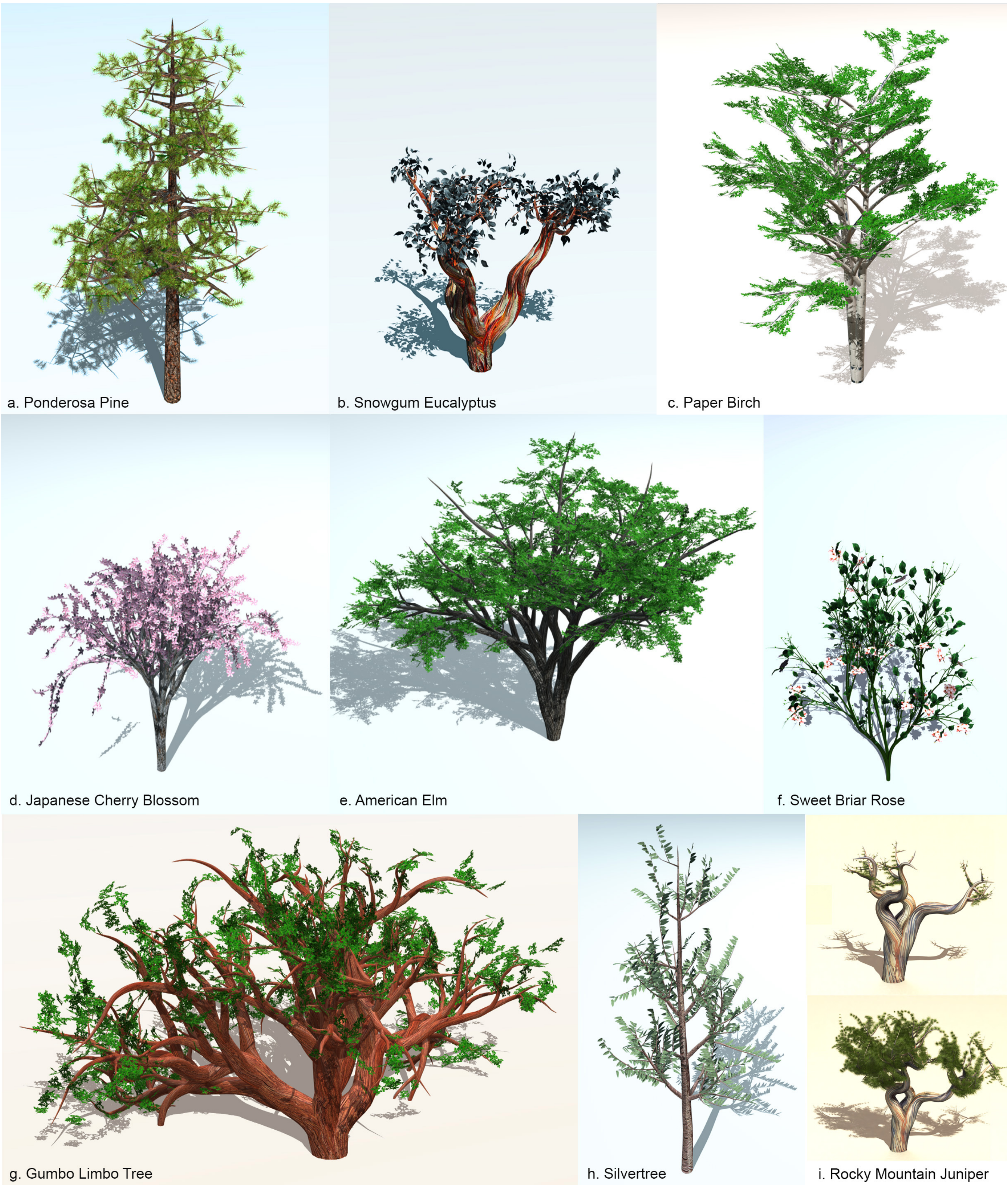

Figure 8: Modeling a variety of common and exotic tree species. a) Ponderosa Pine, *sp. Pinus ponderosa*, b) Snowgum Eucalyptus, *sp. Eucalyptus pauciflora*, c) Paper birch, *sp. Betula papyrifera*, d) Japanese Cherry Blossom, *sp. Prunus serrulata*, e) American Elm, *sp. Ulmus americana*, f) Sweet Briar Rose, *sp. Rosa rubiginosa*, g) Gumbo Limbo, *sp. Bursera simaruba*, h) Silvertree, *sp. Leucadendron argenteum*, i) Rocky Mountain Juniper, *sp. Juniperus scopulorum*

4.1 Appendix A

This section specifies the model parameters in detail. For our experiments the number of branch *spans*, harmonic and linear *terms* were all 4. The maximum levels is typically $3 \leq L \leq 5$.

	General Parameters
S	Tree seed (stateful Mersenne twister)
A	Maximum age, total trunk segments
L	Number of levels, $0 < i < L$
Z	Initial segment size (internodes)
TP_i	Segment taper
TW_i	Segment twist
NS_i	Node spacing
NP_i	Node probability
	Branch Span Parameters
BL_i	Branch lengths, terminal age
BP_i	Branch probability
$s_{k,i}$	Span spacing, $0 < k < spans$
$\phi_{k,i}$	Span angles (see Figures 3a, 4a)
	Branch Linear Parameters
$WS_{k,i}$	Wander spacing, $0 < k < terms$
$WA_{k,i}$	Wander amplitude (see Figure 4b)
WN_i	Wander noise
	Branch Harmonic Parameters
$Hf_{k,i}$	Branch frequency, $0 < k < terms$
$Ha_{k,i}$	Branch amplitude (see Figure 4c)
	Leaf Parameters
LS_i	Leaf spacing
LP_i	Leaf probability
LZ_i	Leaf scale
θ	Leaf orientation
θmin_i	Angle min vector, $\theta min < \theta < \theta max$
θmax_i	Angle max vector

REFERENCES

[1] Jules Bloomenthal. 1985. Modeling the Mighty Maple. *SIGGRAPH Comput. Graph.* 19, 3 (jul 1985), 305–311. https://doi.org/10.1145/325165.325249

[2] Frédéric Boudon, Alexandre Meyer, and Christophe Godin. 2006. Survey on Computer Representations of Trees for Realistic and Efficient Rendering. (2006).

[3] Alenda Chang. 2019. Between Plants and Polygons: SpeedTrees and an Even Speedier History of Digital Morphogenesis. (2019).

[4] Xuejin Chen, Boris Neubert, Ying-Qing Xu, Oliver Deussen, and Sing Bing Kang. 2008. Sketch-Based Tree Modeling Using Markov Random Field. *ACM Trans. Graph.* 27, 5, Article 109 (dec 2008), 9 pages. https://doi.org/10.1145/1409060.1409062

[5] Phillippe de Reffye, Claude Edelin, Jean Françon, Marc Jaeger, and Claude Puech. 1988. Plant Models Faithful to Botanical Structure and Development. *SIGGRAPH Comput. Graph.* 22, 4 (jun 1988), 151–158. https://doi.org/10.1145/378456.378505

[6] Rama Hoetzlein. 2020. Natural Structures through the Convergence of Particles and Shapes. In *Proceedings of the XXII Generative Art Conference.*

[7] M. Holton. 1994. Strands, Gravity and Botanical Tree Imagery. *Computer Graphics Forum* 13 (1994).

[8] Joseph Penn Jason Weber. 1995. Creation and Rendering of Realistic Trees. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95).* Association for Computing Machinery, New York, NY, USA, 119–128. https://doi.org/10.1145/218380.218427

[9] Lina Juuso. 2017. *Procedural Genreation of Imaginative Trees using a Space Colonization Algorithm.* Ph. D. Dissertation.

[10] Brian Karis, Rune Stubbe, and Graham Wihlidal. 2021. A Deep Dive into Unreal Engine's 5 Nanite. *Proceedings of SIGGRAPH 2021* (Aug 2021).

[11] Yoichiro Kawaguchi. 1982. A Morphological Study of the Form of Nature. *SIGGRAPH Comput. Graph.* 16, 3 (jul 1982), 223–232. https://doi.org/10.1145/965145.801284

[12] Sylvain Lefebvre and Fabrice Neyret. 2002. Synthesizing Bark. In *Thirteenth Eurographics Workshop on Rendering.* The Eurographics Association.

[13] Aristid Lindenmayer. 1968. Mathematical models for cellular interactions in development. II. Simple and branching filaments with two-sided inputs. *J Theor Biol* 18, 3 (mar 1968), 300–15. https://doi.org/10.1016/0022-5193(68)90080-5

[14] Bern Lintermann and Oliver Deussen. 1996. Interactive Modelling and Animation of Branching Botanical Structures. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96* (Poitiers, France). Springer-Verlag, Berlin, Heidelberg, 139–151.

[15] Zhihao Liu, Kai Wu, Jianwei Guo, Yunhai Wang, Oliver Deussen, and Zhanglin Cheng. 2021. Single Image Tree Reconstruction via Adversarial Network. *Graphical Models* 117 (2021), 101115. https://doi.org/10.1016/j.gmod.2021.101115

[16] Nelson Max. 1990. Cone-Spheres. *SIGGRAPH Comput. Graph.* 24, 4 (sep 1990), 59–62. https://doi.org/10.1145/97880.97885

[17] Boris Neubert, Thomas Franken, and Oliver Deussen. 2007. Approximate Image-Based Tree-Modeling Using Particle Flows. In *ACM SIGGRAPH 2007 Papers* (San Diego, California) *(SIGGRAPH '07).* Association for Computing Machinery, New York, NY, USA, 88–es. https://doi.org/10.1145/1275808.1276487

[18] Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz. 2009. Self-Organizing Tree Models for Image Synthesis. *ACM Trans. Graph.* 28, 3, Article 58 (jul 2009), 10 pages. https://doi.org/10.1145/1531326.1531364

[19] Joanna L. Power, A. J. Bernheim Brush, Przemyslaw Prusinkiewicz, and David H. Salesin. 1999. Interactive Arrangement of Botanical L-System Models. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (Atlanta, Georgia, USA) *(I3D '99).* Association for Computing Machinery, New York, NY, USA, 175–182. https://doi.org/10.1145/300523.300548

[20] William T. Reeves. 1983. Particle Systems—a Technique for Modeling a Class of Fuzzy Objects. *ACM Trans. Graph.* 2, 2 (apr 1983), 91–108. https://doi.org/10.1145/357318.357320

[21] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. 2007. Modeling Trees with a Space Colonization Algorithm. In *Proceedings of the Third Eurographics Conference on Natural Phenomena* (Prague, Czech Republic) *(NPH'07).* Eurographics Association, Goslar, DEU, 63–70.

[22] Deni Suwardhi, Kamal Nur Fauzan, Agung Budi Harto, Budhy Soeksmantono, Riantini Virtriana, and Arnadi Murtiyoso. 2022. 3D Modeling of Individual Trees from LiDAR and Photogrammetric Point Clouds by Explicit Parametric Representations for Green Open Space (GOS) Management. *ISPRS International Journal of Geo-Information* 11, 3 (2022). https://doi.org/10.3390/ijgi11030174

[23] Ke Xie, Feilong Yan, Andrei Sharf, Oliver Deussen, Hui Huang, and Baoquan Chen. 2016. Tree Modeling with Real Tree-Parts Examples. *IEEE Transactions on Visualization and Computer Graphics* 22, 12 (2016), 2608–2618. https://doi.org/10.1109/TVCG.2015.2513409

[24] Ling Xu and David Mould. 2012. Graphics Interaction: A Procedural Method for Irregular Tree Models. *Comput. Graph.* 36, 8 (dec 2012), 1036–1047. https://doi.org/10.1016/j.cag.2012.08.005

[25] Ling Xu and David Mould. 2015. Procedural Tree Modeling with Guiding Vectors. *Computer Graphics Forum* 34, 7 (oct 2015). https://doi.org/10.1111/cgf.12744