Extending Space Colonization Tree Modeling for Artistic Control and Environmental Interactions

Owen Patrick[1], Manjeet Rege[2] and Reynold Bailey[1]
[1]Department of Computer Science, Rochester Institute of Technology, Rochester, New York, U.S.A.
[2]Department of Graduate Programs in Software, University of St. Thomas, St. Paul, Minnesota, U.S.A.

Keywords: Tree Modeling, Space Colonization.

Abstract: There are a growing number of modeling techniques and algorithms for creating and modifying tree structures, both quickly and realistically. Many of these tools are part of larger software packages used for architecture or landscaping designs. In these tools the tree models are static and come from a library selection. In other areas, such as 3D modeling and design in film and animation, the tree creation tools are more open for artistic creativity and freedom of control. Much research has also been done on growing trees that are sensitive to some environment. That is, growing tree models where genus type, and natural and artificial environment factors are taken into heavy consideration. In this paper, we present an approach that facilitates user creativity while still providing a realistic response to environment factors. Our system adapts the Space Colonization Tree Modeling approach to allow for continued branch addition as well as environmental interaction. Feedback from experienced modelers who participated in a user study revealed that our approach generated the tree skeleton structures they intended and also provided realistic interaction with the environment.

1 INTRODUCTION

Though often overlooked by consumers of most forms of computer generated imagery, botanical tree models are used in many different applications including gaming, movies, urban modeling, and landscape and architecture design. Tree modeling is therefore well-researched in different areas of computer graphics as their creation is not a trivial task in most respects. There are many variations of tree shapes, structures and canopies all of which are controlled or modified by both endogenous information (individual plants genetics) and by exogenous influences (the surrounding environment) (Pirk et al., 2012). These factors must be kept in mind by artists and modelers as different tree systems and algorithms may have limitations on the types of trees and branches that can be simulated. Also the context of the application must be considered as not all instances require a highly detailed or interactive model.

Many existing tree modeling systems have either focused on providing artistic control or on generating realistic response to natural and artificial environment factors. However there has been very little work to bridge the gap between these two approaches. In this paper, we present an approach that facilitates user creativity while still providing a realistic response to environment factors. Our system adapts the Space Colonization Tree Modeling technique (Runions et al., 2007) to allow for continued branch addition as well as environmental interaction. We invited modeling students to use our system and they reported that our approach generated the tree skeleton structures they intended and also provided realistic interaction with the environment.

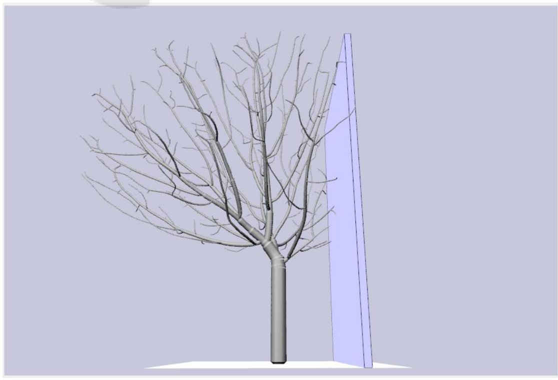

Figure 1: Tree modeled using our modified space colonization technique adapts to the addition of an obstacle to the environment.

The remainder of this paper is organized as fol-

Patrick O., Rege M. and Bailey R..
Extending Space Colonization Tree Modeling for Artistic Control and Environmental Interactions.
DOI: 10.5220/0004690501280135
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (GRAPP-2014), pages 128-135
ISBN: 978-989-758-002-4

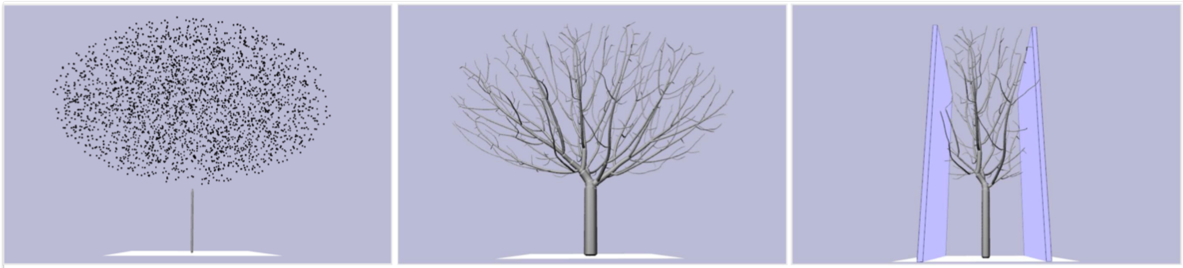

Figure 2: Space Colonization. Initial point cloud in a defined volume (left). Resulting tree model using standard Space Colonization technique (middle). Tree modeled using our modified space colonization technique adapts to the addition of obstacles to the environment (right).

lows: background and related work is presented in section 2, implementation details are presented in section 3, the results of a user study involving experienced modelers is presented in section 4, and the paper concludes in section 5 with a summary of the contributions and potential avenues of future research.

2 BACKGROUND

Direct modeling of trees using NURBS or other parametric techniques allow the user to manually create and position branches until the desired tree structure is attained. Although such approaches offer a high level of artistic control, they are too time consuming to be practical for most applications. To overcome this problem, procedural techniques arose to automatically create less artistically dependent models quickly based on the traversal and application of specific rules that were created and modified by the modeler. These rules controlled things such as branching behavior, branch length, and branch angles (Smith, 1984). Various biological studies of tree structure helped to lay the foundation for these rule-based systems (Lindenmayer, 1968; Honda, 1971; Kawaguchi, 1982; Bloomenthal, 1985; de Reffye et al., 1988).

Several user assisted systems have emerged that attempt to balance the procedural approaches with that of direct modeling. Many of these systems offer the modeler an extensive array of parameters to finely control the characteristics of the tree structure. Weber and Penn (Weber and Penn, 1995), and more recently Deussen and Lintermann (Deussen and Lintermann, 2010), present user assisted tree modeling systems that rely on basic geometric principles rather than trying to adhere to botanical rules. We adopt a similar approach in this work.

To achieve visually accurate models, some researchers have designed techniques that utilize images of trees and tree branch structures. The images can be either registered (Reche-Martinez et al., 2004) or unregistered (Neubert et al., 2007). In both cases, the modeler still has to specify certain parameters for the branches. Other systems combine image-based approaches with rule-based techniques or directly use sketch-based techniques to provide additional control (Chen et al., 2008; Longay et al., 2012).

Researchers have also studied physically-based approaches for tree modeling. Various systems have attempted to simulate the effect of either endogenous influences, exogenous influences, or both. Some of the factors considered in these systems include internal water flow of the plant, the tendency for climbing plants to latch on to structures, the effect of gravity, competition for resources, phototropism, and the presence of obstacles in the environment (Měch and Prusinkiewicz, 1996; Hart and Baker, 1996; Beneš and Millán, 2002; Beneš et al., 2009; Pirk et al., 2012). While these physically-based systems typically produce visually pleasing results and allow for realistic (and in some cases computationally efficient) response to changing environmental factors, they offer limited creative control. In this paper, we adapt the Space Colonization Tree Modeling approach to facilitate user creativity while still providing a realistic response to environment factors

2.1 Space Colonization

The Space Colonization Algorithm for modeling trees (Runions et al., 2007) is a procedural technique that serves as the backbone for our approach. The method itself is much like other procedural algorithms. It uses a simple rule-based approach and mimics the fight for resources that trees and other plants engage in for growth. A volume is first defined in space with a specified number of randomly placed points (see Figure 2). This represents an initial point cloud structure that is placed around a set of starting branch nodes. Each point in the cloud has two primary values; an influence radius i_r and a kill radius k_r. The influence radius defines the effect a given point

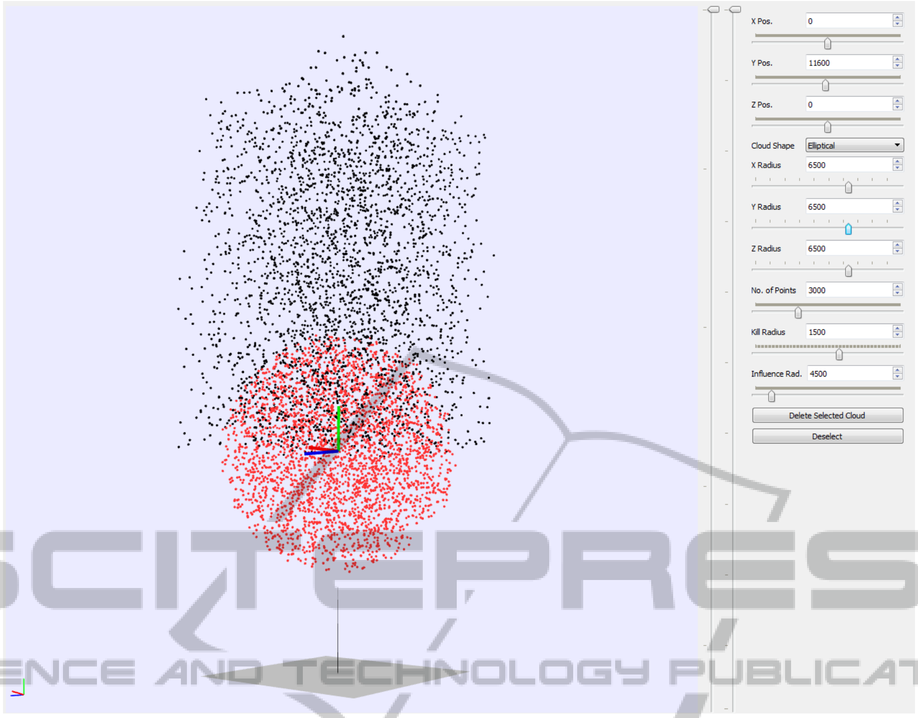

Figure 3: Screenshot of user interface showing two overlapping initial point clouds.

may have on branches nearby. This represents the resource that branches in the system would be attracted to. The kill radius is an area that removes the point from the system once a branch has overlapped with it - representing both the depletion of resources for branch growth and the algorithm's terminating case.

Branches in the system are all of unit length and each has a general grow direction. The algorithm cycles through all branches and computes a direction vector to all points whose influence radii it may fall in. When finished it adds a new branch to the system in that direction. It then checks all new branches against the kill radii of points still in the system. Once a branch enters the kill radius of a point, that point is removed from the point cloud. This process is repeated until there are no more points in the cloud. The technique produces attractive and, in general, physically plausible results.

Pirk et. al. (Pirk et al., 2012) noted that one of the drawbacks of the Space Colonization technique is the need to regrow the entire tree structure to adapt to changing external factors. In their approach, they instead take a fully formed tree skeleton as the base model and compute its method of growth. In doing this they have the total span of branches that could possibly be created and so can adapt the tree to varying surroundings. Their system provides visually pleasing results, however their system is still locked into the set of branches that are initially provided and adding more is not possible. Furthermore, the precomputation required for their models is quite expensive. In our approach we augment the space colonization technique by way of parametrization and present it in context for user assisted modeling. This enables a greater amount of artistic freedom while still providing realistic response to the surrounding environment.

3 IMPLEMENTATION DETAILS

We implemented our system in two different environments to allow usability testing on both a portable laptop and a stand-alone desktop. The specification of each system is shown in the table below:

	Laptop	*Desktop*
OS	Ubuntu 12.04	Windows 8
CPU	Intel Core i5 (2.50 GHz)	Intel Core i5 (3.8GHz)
GPU	Nvidia GeForce GT 540M	Nvidia GeForce GTX 660 Ti
RAM	4GB DDR3	8GB DDR3

To ensure cross-platform capabilities, we use Qt IDE - Qt Creator for user-interface development. Our software is written in C++ and utilizes OpenGL as the rendering agent.

3.1 Approach

Our approach can be split into three main processes:

1. Growing the tree model using the space colonization approach proposed by Runions et al. (Runions et al., 2007).
2. Ensuring the model conforms to external physical factors. To accomplish this we emulate the approach proposed by Pirk et al. (Pirk et al., 2012).
3. Providing a mechanism for continuing tree growth based on user input. The additional growth also conforms to the external physical factors.

The user interface provides the modeler with several options to create point clouds of varying sizes based on four basic shapes; sphere, ellipse, cone, and cube. The modeler can create any number of starting point clouds and position them in the scene as needed (see Figure 3). Each cloud will have its own influence and kill radius applied if needed. These point clouds are placed around an initial set of branches that represent the starting trunk (or root) of the branch system. For this project, we have implemented an initial trunk that grows perpendicular to the ground-plane. The user has the ability to modify the width and number of starting branches (recall that all branches are unit length). An initial point cloud and starting trunk setup is shown in Figure 2 on the left and the resulting tree structure is shown in the middle. The user can then specify different obstacles in the scene or chose to add on to current set of branches by specifying additional point clouds. This process can be repeated until the user is satisfied with the result.

The space colonization growth process is computationally intensive and even with preprocessing to speed this up, the modeler has no way of predicting what the final tree structure will be beyond the general space the branches will occupy (based on the point cloud they specify). To overcome this problem and provide finer artistic control, we extend the basic space colonization technique by providing a parameter called *step size* which allows the user to specify how many nodes should be processed before pausing the growth algorithm. The modeler can set small step sizes to periodically pause and preview the tree's growth. They can also modify, remove, or add new point cloud information or change kill radius or influence radius parameters to affect the subsequent growth of the branches. Note that since we maintain age information of the branches that the modeler can also reverse branch-growth to return to an earlier stage of the tree where they can also make changes.

For every iteration of growth (i.e. whenever new branches are added to the existing tree) we cycle through the tree, starting from the branch tips (any branches without children) and add girth going towards the root node. This girth is changed at each age step and also when the user dynamically alters the tree's age, environment, or the girth value itself.

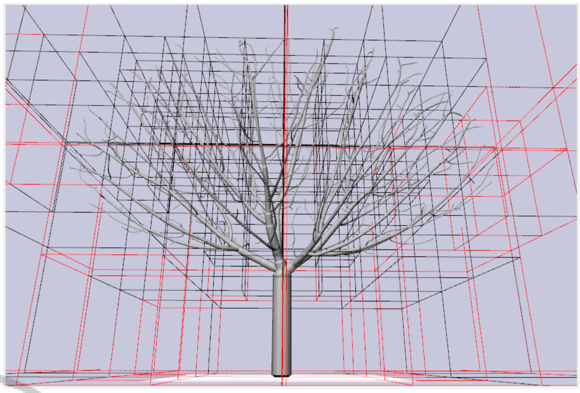

Figure 4: Octree spatial data structure used to speed up collision testing between branches and obstacles in the environment.

At any step of the growth process, the modeler can add obstacles to the scene. The addition (or modification) of obstacles triggers an intersection test that checks each branch in the tree against each obstacle in the scene. Furthermore, once the obstacle is in place, the test must be conducted for each subsequent iteration. To speed up this process, we use an octree spatial data structure to store and retrieve branch position information as shown in Figure 4. If a branch is found to be in a space occupied by an obstacle, an iterative function is called which systematically disables that branch and all children it may have, the results of which can be seen in Figure 2 (right).

3.2 Gravitropism

In our system, we have implemented an option for a loose variation of gravitropism where the gravity influences the direction of branch growth. Unlike the work in TreeSketch (Longay et al., 2012), our implementation does not modify existing branches as this requires that the complete tree be regrown. Of course we can go back in time and modify an earlier stage of the growth process if that is the effect desired by the artist. Our gravitropism effect is accomplished by further modifying the basic space colonization method to grow new branches by taking in an additional weighted direction vector to account for the addition of gravity. The weight applied to each branch is scaled by the distance from the current branch node to the center of the branch system. This means that the further a branch is from the tree center, the more it will bend towards the ground plane. A more elab-

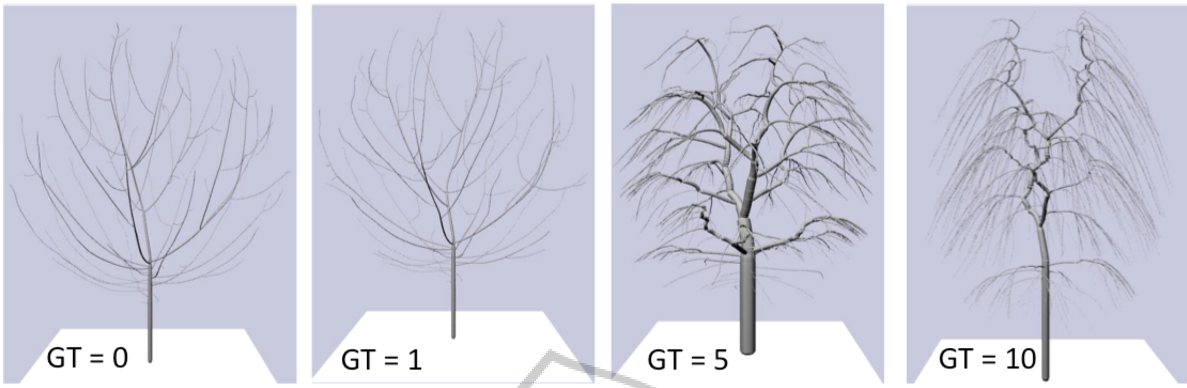

Figure 5: Trees modeled with different amounts of gravitropism weights. As seen on the right, large weights can result in unnatural looking tree structures.

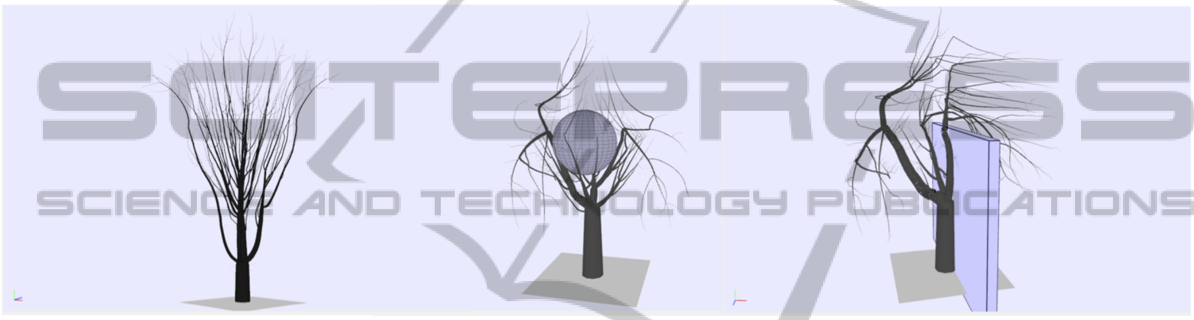

Figure 6: Examples illustrating the level of control available to the artist with our system.

orate method could also be implemented to take into account the summed weight of the children branches supported by the current branch node. Figure 5 shows some results using different amounts of gravitropism weights. Note that large weights can result in unrealistic results however this is completely within the artist's control.

3.3 Artist control

As described above, the artist has the ability to pause growth at any time and to modify the point cloud and other parameters to influence the future growth of the tree. With this level of control, the artist can of course create unnatural looking trees. However, regardless of how extreme the resulting models are, they will still react in an appropriate manner to obstacles in the environment. Figure 6 shows some examples which illustrate the level of control that is available to the artist.

Figure 7 shows how our tool can be used to generate tree structures that are visually similar to real world scenes. The photographs on the left show the front and side view of a tree between two bike stations. We did a rough mockup of the bike stations in Maya and left space for the tree in the middle. The Maya scene was imported into our tool which was then used to generate the tree model to fill the space.

4 USER STUDY

We conducted a user study in order to assess the usefulness of our tool in the workflow of experienced modelers. Six modelers (4 males, 2 females) with at least one year experience with packages such as Maya and Maxon Cinema 4D volunteered to participate in our study.

For each participant, the first 10 minutes of the study were for an introduction and tutorial about our system. They then spent 20 minutes using the tool and 5 minutes completing a survey. During the 20-minute session, they were tasked with setting up a scene with a tree and one or more obstacles. We instructed the subjects to only consider tree structure and to disregard lighting/shading and other visual effects as that was not the focus of our work.

The survey consisted of four statements and the responses were based on a Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree):

S1. The structure of botanical trees affects the visual

Figure 7: Using our tool to generate trees that have the same structure and response to the environment as real-world trees. Photographs on left show front and side view of tree between two bike stations. Images on the right show corresponding rendered views.

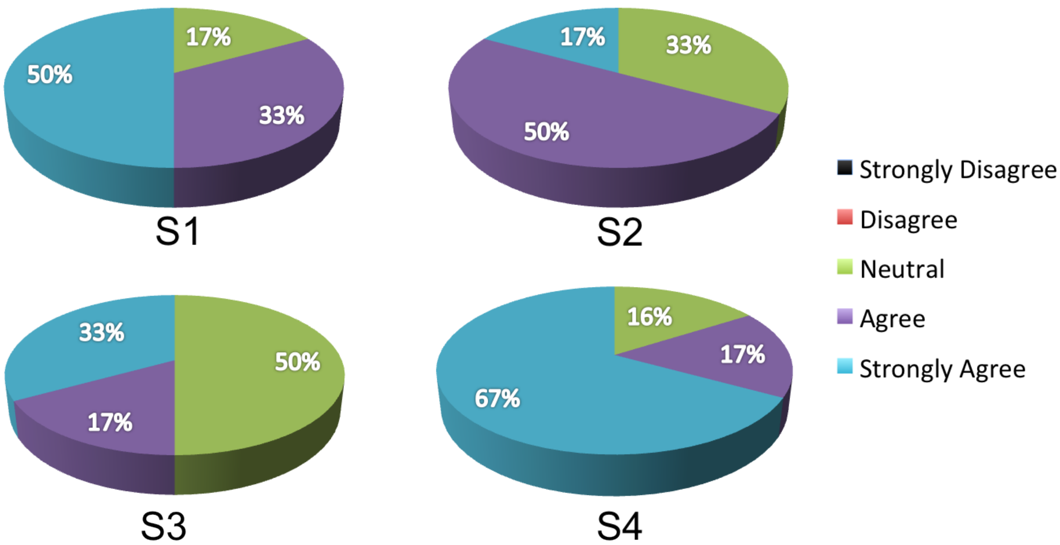

Figure 8: Results of survey completed by six experienced modelers.

appeal of digital imagery.
S2. When obstacles are present, the resulting tree shape appears realistic.
S3. The controls provided allow you to create the tree structure you intended.
S4. I would like to see this tree modeling approach become part of the modeler's work flow.

The results of the questionnaire are shown in Figure 8. Note that none of the subjects disagreed or strongly disagreed with any of the statements. A notable observation from the survey results is that 67% strongly agreed that our approach to tree modeling should be part of the modeler's workflow.

5 CONCLUSIONS

In this paper we presented a system that adapts the Space Colonization Tree Modeling approach to allow for continued branch addition as well as environmental interaction. Our approach facilitates user creativity while still providing physically plausible models. The results of a user study with experienced modelers demonstrates the usefulness of our system.

Our system does have several limitations which we hope to address as part of our future work. Our gravitropism implementation, for example, only works well if the initial point cloud is roughly centered above the starting trunk. Unrealistically leaning trees can result if this is not the case as our gravitropism computation for each branch is based on distance from the trunk. Additional limitations include the fact that our system does not yet consider lighting (phototropism) or curved trunks. We also plan to further develop our data-structures for branch control. A branch container would allow us the ability to implement our own branch clustering algorithms for interactivity and subsequent balancing strategies. Finally, we would like to expand this tool to operate on the GPU to give faster responses and open more doors for real-time computation.

REFERENCES

Beneš, B., Andrysco, N., and Šťava, O. (2009). Interactive modeling of virtual ecosystems. In *Proceedings of the Fifth Eurographics conference on Natural Phenomena*, NPH'09, pages 9–16, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Beneš, B. and Millán, E. U. (2002). Virtual climbing plants competing for space. In *Proceedings of the Computer Animation*, CA '02, pages 33–, Washington, DC, USA. IEEE Computer Society.

Bloomenthal, J. (1985). Modeling the mighty maple. *SIGGRAPH Comput. Graph.*, 19(3):305–311.

Chen, X., Neubert, B., Xu, Y.-Q., Deussen, O., and Kang, S. B. (2008). Sketch-based tree modeling using markov random field. *ACM Trans. Graph.*, 27(5):109:1–109:9.

de Reffye, P., Edelin, C., Françon, J., Jaeger, M., and Puech, C. (1988). Plant models faithful to botanical structure and development. *SIGGRAPH Comput. Graph.*, 22(4):151–158.

Deussen, O. and Lintermann, B. (2010). *Digital Design of Nature: Computer Generated Plants and Organics*. Springer Publishing Company, Incorporated, 1st edition.

Hart, J. C. and Baker, B. (1996). Structural simulation of tree growth and response. In *In: Proceedings International Conference on Shape Modeling and Applications*, pages 7–11. Springer-Verlag.

Honda, H. (1971). 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*, 31(2):331–338.

Kawaguchi, Y. (1982). A morphological study of the form of nature. *SIGGRAPH Comput. Graph.*, 16(3):223–232.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development: Parts I and II. *Journal of Theoretical Biology*, 18.

Longay, S., Runions, A., Boudon, F., and Prusinkiewicz, P. (2012). Treesketch: interactive procedural modeling of trees on a tablet. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, pages 107–120, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Měch, R. and Prusinkiewicz, P. (1996). Visual models of plants interacting with their environment. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 397–410, New York, NY, USA. ACM.

Neubert, B., Franken, T., and Deussen, O. (2007). Approximate image-based tree-modeling using particle flows. *ACM Trans. Graph.*, 26(3).

Pirk, S., Stava, O., Kratt, J., Said, M. A. M., Neubert, B., Měch, R., Benes, B., and Deussen, O. (2012). Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. Graph.*, 31(4):50:1–50:10.

Reche-Martinez, A., Martin, I., and Drettakis, G. (2004). Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Trans. Graph.*, 23(3):720–727.

Runions, A., Lane, B., and Prusinkiewicz, P. (2007). Modeling trees with a space colonization algorithm. In *Proceedings of the Third Eurographics conference on Natural Phenomena*, NPH'07, pages 63–70, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Smith, A. R. (1984). Plants, fractals, and formal languages. *SIGGRAPH Comput. Graph.*, 18(3):1–10.

Weber, J. and Penn, J. (1995). Creation and rendering of realistic trees. In *Proceedings of the 22nd annual con-*

ference on Computer graphics and interactive techniques, SIGGRAPH '95, pages 119–128, New York, NY, USA. ACM.