Fast Estimation of Growth Direction in Virtual Plants Simulation

Bedřich Beneš

Department of Computer Science, Czech Technical University,
Karlovo nám. 13, Prague, Czech Rep.
e-mail: benes@sgi.felk.cvut.cz

From: Nadia Magnenat-Thalmann and Václav Skala editors,
The Fifth International Conference In Central Europe on Computer Graphics and Visualization – WSCG'97
University of West Bohemia Press, 1997, Vol: I; pp: 1–10

Fast Estimation of Growth Direction in Virtual Plants Simulation

Bedřich Beneš
Department of Computer Science, Czech Technical University,
Karlovo nám. 13, Prague, Czech Rep.
e–mail: benes@sgi.felk.cvut.cz

February 20, 1997

Abstract
This paper deals with visual models of plants. A new method for fast determination of growth direction of the growing elements in artificial plants is presented here. The most important factor is the amount and direction of light incidenting surface of leaves in our simulation. We also take into account fluttering of the leaves in the wind. The method for the light calculation presented here is a significant extension of the algorithm presented in [1]. There are two important properties of the method which is based on sampling. The number of samples is independent to the number of elements in scene and every shadow in the scene is considered - not only shadows of the leaves but also shadows casted by the branches, fruits, etc.

1 Introduction

From the beginning of computer graphics, visual models of plants have attracted the interest of researchers. The first aim was just to obtain realistic visual models of plants as fast as possible without considering behaviour of the plants in their environment. However, some of the shapes can be achieved easily (or maybe only?) by means of biological simulation of the plant development with respect to the environment affecting the plant. This fact has been considered and consequently overgrew into simulations of plants interacting with their environment several years ago. The results seem to be very promising for the future research in area of computer graphics as well as in biology, ecology, etc.

We introduce significantly improved method for light estimation in simulation of plant development based on work [1]. We do not deal neither with the precise geometry of the plant (*cf.* [2, 10, 19]) nor with formal specification of the plant development [14]. We use biologically based stochastic model of plant architecture and development according to de Reffye *et al.* [4] (see Figure 1).

2 Previous work

Voxel space automata has been used in computer graphics to model plants which are strongly affected by the environment. Greene [7] used voxel space automata to simulate growth of climbing plants and to simulate roots searching their paths on stony ground [6]. Stochastic growth (a random walk) is used in his simulations. Multiple random trials are made, position of a growing element and its orientation are randomly perturbed, fitness function is evaluated, and the best position and orientation are used as the new growth direction. The distance from the center of mass is evaluated for every element. If the trial closest to the obstacle is used, the effect of climbing plants is very easily achieved in this simulation. The amount of light is calculated by 3DDA sampling of the trajectory of the sun. Several rays are cast for every growth element and the coefficient of sky exposure is evaluated as relative number of occluded and free rays. The effect of heliotropism is achieved by constructing the illumination table for each trial. Several rays are cast to cover as much from the sky as possible. Black table entries represent obstructed rays and white unobstructed ones. A low pass filter is then used and the brightest spot on the sky is found. This place corresponds to the direction of further growth of the growing element. Apparently number of the casted rays depends on the number of nodes and the precision of the simulation.

An alternative method of the light direction estimation is presented in the work of Chiba *et al.* [8]. They assume that the amount of light is constant for every point on the celestial sphere. They introduce a *leaf-ball* which is an approximation of a cluster of leaves. This object is translucent with transparency according to number of included leaves. The light estimation is determined by projection of these balls on the celestial sphere. The light is estimated using the hidden surface algorithm or 3DDA in voxel space. The brightest direction is defined by summing the participating vectors associated with each ray reaching the celestial sphere.

The theory of parallel string rewriting systems introduced by Lindenmayer [9] plays the leading role in visual modeling of plants at present. Smith [18] applied the Lindenmayer's concept to generate models of artificial plants and named this method *L-system*. The concept of a string generation by rewriting and its geometrical interpretation by turtle graphics is well known in computer graphics and we do not deal with this in general here. The survey of L-systems is presented in the book [15] and the latest work can be found in [12, 14].

There are two important environmentally sensitive extensions of L-systems published [12, 13]. The context sensitive parametric L-systems was extended by so called query modules [13]. The query modules are parametrized components of the rewriting process and they can read some values from the surrounding environment; *e.g.*, distance from the obstacle. Parameters of the module are set when the rewriting process asks the query module for them. Drawback of this concept is that the whole plant in the string rewriting phase has to be always constructed.

Open L-systems [12] do not suffer from this drawback. The rewriting step is preceded with so called *scanning phase*, in which every so called *communication module* has assigned value of every parameter without geometrical interpretation of the string. The plant and the environment are treated as processes communicating together via messages. The communication modules can ask the environment for some values (*e.g.*, presence of the obstacle or amount of the light) and they can also inform the environment of some values (*e.g.*, carbon oxide).

This theoretical framework is very important although only few algorithms for calculation of such interaction with environment in respect with plants have been published. In [12], the plant is encompassed by grid of voxels in resolution of $32 \times 32 \times 32$. The voxels are allocated by translucent clusters of leaves L_i. The sky hemisphere is approximated with (in [12] nine) directional light sources E and raycasting is performed from the center of every cluster L_i to the light sources. Every cluster of leaves L_j which is hit by the ray casted from L_i is then projected to the plane perpendicular to the direction of light source E as well as the cluster L_i from which the rays are casted. Then intersecting areas A_{ij} of projected clusters L_j and raycasting cluster L_i are computed and total light flux is computed using difference of total projected area A_i of L_i and intersecting area A_{ij}. Because the leaf clusters are treated as translucent, the total flux coming from light source E is increased by proportion of flux coming through the intersection A_{ij} to the cluster of leaves L_i.

This approach is efficient, but suffers from certain oversimplifications. First one is small number of light sources. Using small number of light sources would probably cause the branches to grow in distinct directions, greater number of lights will cause increasing time of simulation. Another simplification is the low number of clusters causing again errors in simulation. Increasing this number causes higher execution time of simulation. Apparently the number of rays casted in the simulation will increase according to the number of leaves.

3 The growth model

We use the stochastic model according to de Reffye *et al.* [4] (see Figure 1a)). The

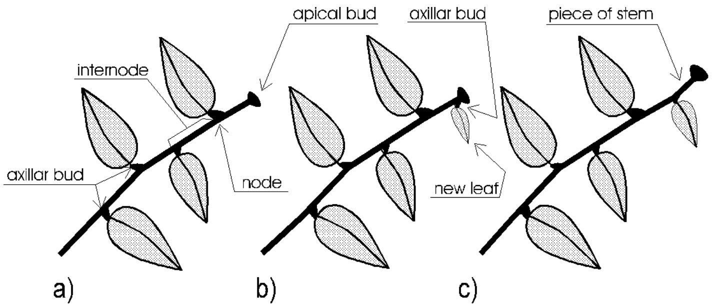

Figure 1: A branch model according to de Reffye *et al.*. a) Branch model with description of the elements of plant. b) Apex produces the axillar leaf and axillar bud appears at the leaf's axil. c) Apical bud generates internode stem.

fundamental elements of this model are *bud*, *leaf*, and *stem*. The bud is a basic growing element which can perform several actions. The actions depend on external conditions (light, water, nutrients, etc.) as well as on internal ones (age, amount of meristem, etc.) of the plant. A bud can either die, or bloom and die, or sleep for a while, or become an internode. Process of becoming an internode (*cf.* Figure 1) is the most important action and allows simulation of growing and branching. This process consist of three steps. In the first step, one or more so called axillary leaves appear beside the bud (*cf.* Figure 1b)) and new bud(s) appear(s) at their axils. Then the bud produces piece of stem - the new internode (Figure 1c)). The apical bud is called *the apex*.

4 Estimation of the light flux

Estimation of the light flux from the sky hemisphere reaching every leaf consists of several distinct steps. First the sky hemisphere is approximated by polyhedron and the flux densities (radiosities) are evaluated for its faces. Then the sampling of the plant from these faces is performed, contributions of the the flux for leaves are computed, and the total light flux for every leaf (irradiance [3]) is evaluated.

4.1 Approximation of the sky hemisphere

In the first step, the sky hemisphere is approximated by polyhedron consisting of four-sided faces of the same (unit) area (see Figure 2).

Let p denote number of the faces and let $F_k;\ k = 1, ..., p$ denote these faces. The flux

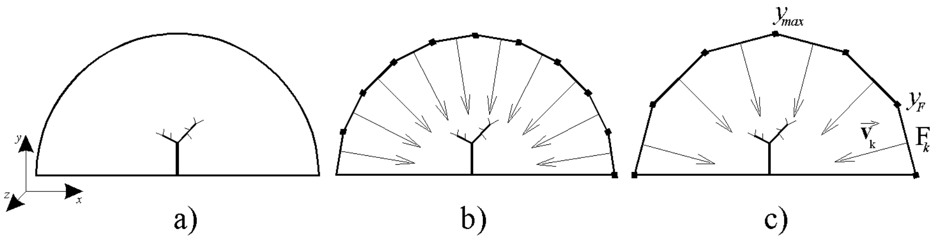

Figure 2: Projection of the sky hemisphere to the xy plane and its two approximations with sampling directions $\mathbf{v}_k$ displayed.

density of the face F_k is denoted by B_k and corresponds with the amount of light coming to the scene from area of sky, which is approximated by the face F_k. The flux density B_k is computed as follows

$$B_k = \bar{y}\ B_c + \begin{cases} \bar{y}\ B_s; & \text{if the sun visits this face,} \\ 0; & \text{otherwise.} \end{cases} \quad (1)$$

where B_c represents constant amount of flux density (recall here, that the faces F_k have equal area) and B_s denotes contribution of the sun. Coefficient $\bar{y}$ is calculated as

$$\bar{y} = y_F / y_{max},$$

where y_{max} is the y coordinate of the top of the hemisphere and y_F is maximal y coordinate of the face F_k (see Figure 2c)). We recall here that $0 \leq \bar{y} \leq 1$. Coefficient $\bar{y}$ is greater for faces near to the top of the sky hemisphere which simulates the fact that the amount of light coming in the noon is higher [3].

The equation (1) also reflects the trajectory of the sun. The flux density of the faces which are "visited" by sun is increased by factor $\bar{y}B_s$.

Once the flux densities B_k are computed, the direction vector denoted by $\mathbf{v}_k$ is assigned to every face F_k (see Figure 2). This vector is perpendicular to its face and points to the plant.

4.2 Sampling

Next is a sampling phase. We compute the light flux Φ coming from the sky to *some* leaves on the plant. Not all, but only several leaves (typically the axillar ones) are important to further growth direction of bud.

Let L_i denote the cluster of leaves, where $i = 1, ..., n$ and n is the number of buds. All leaves in cluster L_i important for i-th bud have assigned unique color denoted C_i. All the others objects in the scene have the same color as the background, so they are rendered and therefore they cast shadows. In contradiction to [8] and [12] *every* object casts shadows in our simulation.

When the colors are assigned to the clusters of leaves, the plant is rendered using Z-buffer algorithm, constant shading, and parallel projection [5]. The camera is placed into the middle of the face F_k and its viewing vector is set to the vector $\mathbf{v}_k$ (we simply look from the hemisphere to the plant, see Figure 3).

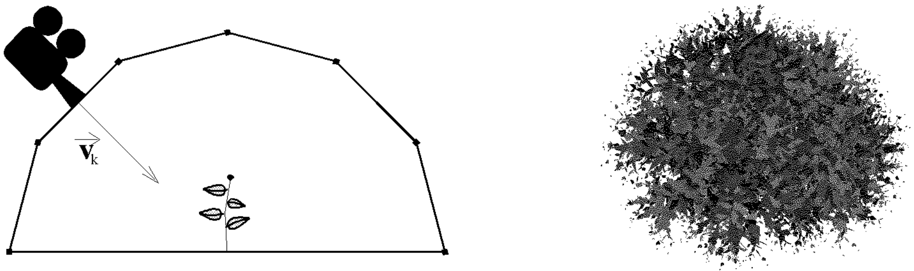

Figure 3: Sampling of the plant and corresponding view.

Now we will look in detail how to compute the light flux denoted by $\Phi_{k,i}$, which is coming from the face F_k to cluster of leaves L_i. First, we compute the area of the leaves which is visible from the sampling direction $\mathbf{v}_k$. We use sigmoidal growth function [15] so the relationship between the age and the area of the leaf is known. Suppose cluster of leaves L_i consists of m leaves and denote their area by a_j $j = 1, ..., m$. The area of L_i is denoted by A_i and

$$A_i = \sum_{j=1}^{m} a_j. \tag{2}$$

Suppose all leaves in cluster L_i are planar and they lay in the plane perpendicular to vector $\mathbf{v}_k$. Suppose all these leaves are visible. Using rendering resolution $\mathrm{res}_x \times \mathrm{res}_y$ and camera viewport $\mathrm{orho}_x \times \mathrm{ortho}_y$, because *parallel projection* is used, we will see at most H_i pixels from the cluster L_i

$$H_i = \frac{\mathrm{res}_x \mathrm{res}_y}{\mathrm{ortho}_x \mathrm{ortho}_y} A_i = c\ A_i. \tag{3}$$

We compute the *histogram* (vector of absolute frequencies) of colors C for the k-th sample. Let $H^e_{i,k}$ denote the value obtained from histogram for the L_i in k-th sample. The relative number of exposure $S_{i,k}$ of the L_i can be easily evaluated from (3) and the histogram value $H^e_{i,k}$ as

$$S_{i,k} = \frac{H^e_{i,k}}{H_i}. \tag{4}$$

Obviously $0 \leq S_{i,k} \leq 1$. Value zero corresponds to the fact that the cluster of leaves L_i lays in the shadow.

The effective area of L_i affected by the flux coming from face F_k is denoted by $A^e_{i,k}$

$$A^e_{i,k} = S_{i,k}\ A_i. \tag{5}$$

We see from (3) and (4) that

$$A^e_{i,k} = \frac{1}{c} H^e_{i,k}. \tag{6}$$

So the light flux coming from face F_k to L_i is

$$\Phi_{i,k} = A^e_{i,k}\ B_k \tag{7}$$

The total light flux Φ_i from the whole hemisphere to the cluster of leaves L_i is

$$\Phi_i = \sum_{k=1}^{p} \Phi_{k.i} \tag{8}$$

The total light flux coming to clusters of leaves is increased by the amount of ambient light denoted by Φ_a, presented in the crown of the tree. In our simulation, we use 10% of total light flux of the hemisphere

$$\Phi_a = 0.1\Phi.$$

We use constant amount of ambient light in the crown, although we know that Reeves and Blau [16] use exponential scaling of this coefficient in their simulation. It does not change the shape of the plant significantly if directional light fluxes are considered.

4.3 Oscillation of leaves

One of the disadvantages of this kind of sampling is that planar leaves sampled in the direction perpendicular to their normal are invisible, so the leaves are treated as completely hidden. In order to solve this problem we consider leaves as in the long term simulation – we include local oscillations of leaves.

First we compute the main axis and the adjoining axis as indicated on the Figure 4a). Then we approximate the leaf with an ellipsoid (the Figure 4b)).

One objection to this method could be that we enlarge the effective area of leaves with this approximation. But we do not need to know the exact light flux. We need to identify the direction of the utmost light flux in order to decide where buds will grow to. For this evaluation the exact amount of the light is unimportant.

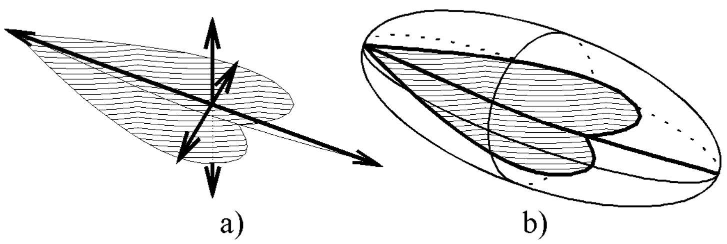

Figure 4: Bounding ellipsoid is the approximation of the leaf.

5 Evaluation of the growth direction

Recall that the vector $\mathbf{v}_k$ is the direction where the light flux $\Phi_{k,i}$ comes from to the cluster of leaves L_i. The direction to the brightest spot is denoted by $\mathbf{v}_k^{max}$ and is opposite to vector $\mathbf{v}_k$ of maximal flux

$$max\{\Phi_{k,i}; k = 1, ..., p\}.$$

If there are more equal light fluxes, one of them is randomly chosen [1].

Let h_i denote the growth direction of i-th bud. The direction to the brightest spot modifies this growth direction as follows (see Figure 5)

$$\mathbf{h}'_i = (1 - H)\mathbf{h}_i - H\mathbf{v}_k^{max}, \tag{9}$$

where $\mathbf{h}'_i$ is the new growth direction, $-\mathbf{v}_k^{max}$ is the direction to the brightest spot and $0 \le H \le 1$ is coefficient of *heliotropism* or *sun seeking*.

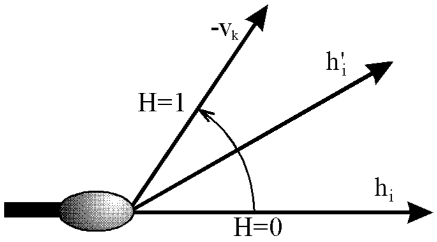

Figure 5: Calculation of new growth direction h'_i of a bud.

6 Implementation and results

We use Silicon Graphics Indigo^2 with Extreme graphics board and R4400 on 200MHz. We use hardware *Z-buffer* for the light estimation. It takes approximately 30 minutes to generate a sequence consisting of 350 script files for the `POVRay` raytracer with about 250 buds and 50 samples per frame. A sampling area is $\mathrm{res}_x \times \mathrm{res}_y = 400 \times 400$ pixels.

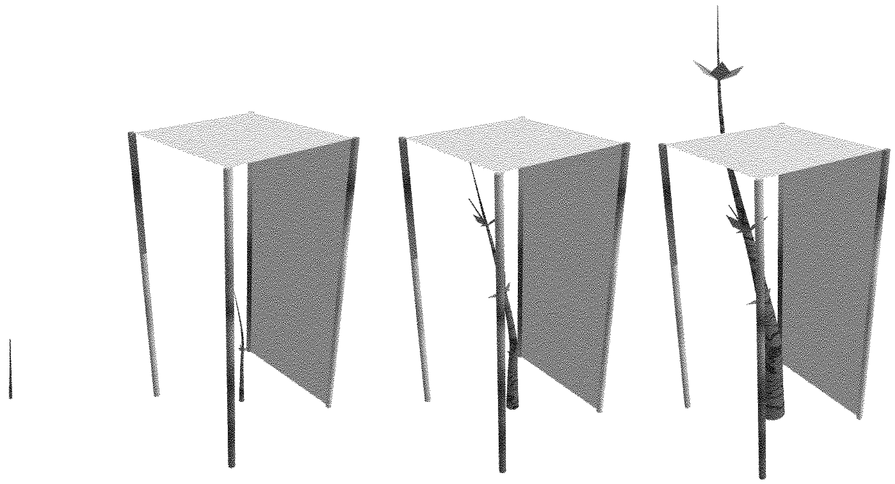

Figure 6: Several frames from animation of a synthetical stem development affected with a light obstacle.

Figures 6 and 7 show plants growing around the obstacles (no collision detection has been solved here). The animation sequence consists of 10 frames and was generated in time less than 20 seconds.

This method has the same aliasing problem as every method based on sampling and reading of the contents of Z-buffer [11] or [17]. The simple (but unefficient) solution to this problem is enlargement of the sampling area resolution. More efficient solution of this problem is a topic of our current research.

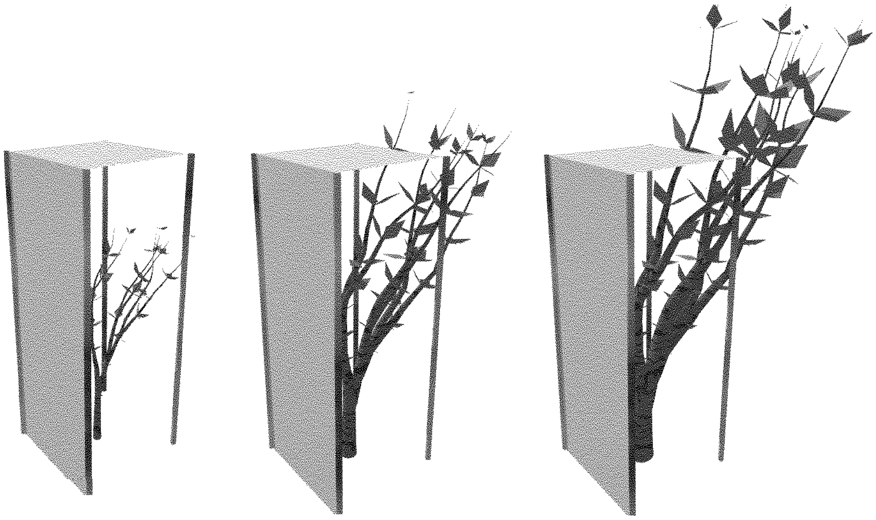

Figure 7: Three frames from animation of the artificial plant development under influence of the light with the obstacle.

7 Conclusions

A new method for fast estimation of growth direction in simulation of plant development has been described here. The method estimates light in constant number of samples of the plant and considers local shadows casted by leaves as well as all of the others parts of the plant. This method also includes fluttering of leaves.

Animations are located on Web page `http://sgi.felk.cvut.cz/~ benes`.

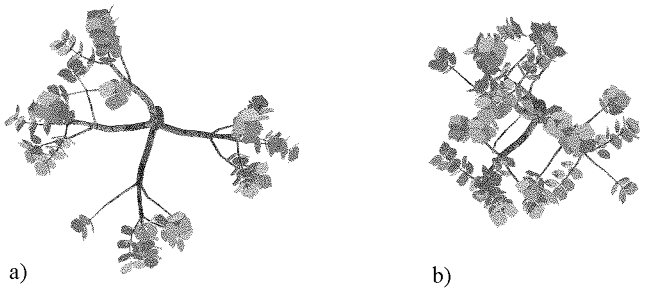

Figure 8: a) branches do not compete and b) compete for the light coming from the top.

8 Acknowledgements

We would like to thank Pavel Tvrdík for a full re-reading of the paper.

References

[1] B. Beneš. An Efficient Estimation of Light in Simulation of Plant Development. In R.Boulic and G.Hégron, editors, *Computer Animation and Simulation'96*, Springer Computer Science, pages 153–165. Springer-Verlag Wien New York, 1996.

[2] J. Bloomenthal. Modeling the mighty maple. In *Proceedings of SIGGRAPH '85*, volume 19(3), pages 305–311, 1985.

[3] M. Cohen and J. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Prosfessional, 1993.

[4] P. de Reffye, C. Edelin, J. Fracon, M. Jaeger, and C. Puech. Plants models faithful to botanical structure and developement. In *Proceedings of SIGGRAPH '88*, volume 22(4), pages 151–158, 1988.

[5] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, 1990.

[6] N. Greene. Detailing tree skeleton with voxel automata. *SIGGRAPH'91, Course Notes on Photorealistic Volume Modeling and Rendering Techniques*, 1991.

[7] N. Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. In *Proceedings of SIGGRAPH '89*, volume 23(4), pages 175–184, 1989.

[8] N. Chiba, S. Okawa, K. Muroaka, and M. Muira. Visual simulation of botanical trees based on virtual heliotropism and dormancy break. *The Journal of Visualization and Computer Animation*, 5:3–15, 1994.

[9] A. Lindenmayer. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, Parts I and II(18):280–315, 1968.

[10] B. Lintermann and O. Deusen. Interactive Modelling and Animation of Branchinf Botanical Structures. In R.Boulic and G.Hégron, editors, *Computer Animation and Simulation'96*, Springer Computer Science, pages 139–151. Springer-Verlag Wien New York, 1996.

[11] N. Max. Hierarchical Rendering of Trees from Precomputed Multi-Layer Z-Buffers. In *Rendering Techniques'96*, Springer Computer Science, pages 165–174. Springer-Verlag Wien New York, 1996.

[12] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *Proceedings of SIGGRAPH '96*, volume 30(4), pages 397–410, 1996.

[13] P. Prusinkiewicz, M. James, and R. Měch. Synthetic Topiary. In *Proceedings of SIGGRAPH '94*, volume I, pages 351–358, 1993.

[14] P. Prusinkiewicz, M. James, R. Měch, and J. Hannan. The Artificial Life of Plants. In *Course Notes of SIGGRAPH '95*, volume I, pages 1–38, 1995.

[15] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag, New York, 1990. With J.S.Hannan, F.D. Fracchia, D.R.Fowler, M.J.de Boer, and L.Mercer.

[16] W. Reeves and R. Blau. Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems. In *Proceedings of SIGGRAPH '85*, volume 19(3), pages 313–322, 1985.

[17] W. Reeves, D. Salesin, and R. Cook. Rendering Antialiased Shadows with Depth Maps. In *Proceedings of SIGGRAPH '87*, volume 21(4), pages 283–291, 1987.

[18] A. Smith. Plants, fractals and formal languages. In *Proceedings of SIGGRAPH '84*, volume 18(3), pages 1–10, 1984.

[19] J. Weber and J. Penn. Creation and Rendering of Realistic Trees. In *Proceedings of SIGGRAPH '95*, volume 22(4). SIGGRAPH New York, 1995.