An Efficient Estimation of Light in Simulation of Plant Development

Bedřich Beneš

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

From: Ronan Boulic and Gerard Hégron editors,
Computer Animation and Simulation'96
Springer–Verlag Wien New York, 1996, pp: 153–165

An Efficient Estimation of Light in Simulation of Plant Development

Bedřich Beneš

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

Abstract. During a simulation of plant development evaluation of the amount of light plays a significant role. Most of the previous works take into account only constant amount and constant direction of the light in the scene without respect to local shadows in the tree. In the other works this evaluation strongly depends on the number of objects in the scene.

This paper introduces a new method for evaluation of amount of light for the artificial plants. This technique is based on Z-buffer algorithm and has the ability to evaluate the direction and the amount of light for every leaf in the plant with a significantly decreased time of calculation. Several new aspects of lifetime of the plant elements and of the whole plant are discussed.

1 Introduction

Plenty of papers have been written about the plants in the computer graphics and also many classifications of published methods have been presented. One classification would be done from the view of the aim of the method. Some of the methods are useful for particular reason e.g for modeling of the the phylotaxis [4], for generating plants in real time [10, 22], for the film industry [17]. The others use formal methods and they try to describe some model or method in general regardless to the application which is found subsequently. Some of these methods generate climbing plants in voxel space [5], some use strands topology for description of branching [6], some use string rewriting systems for simulation of plant development [14] or use a combinatorial approach for describing the branching structure [21].

Another possible classification introduced by [19] focuses the generated structure. The model is said to be *topological* or *structural* oriented if the generated structure carries information mostly about the adjacency of the parts [6, 14, 19] or [21]. On the other hand, the model is *geometrically* or *space* oriented if it primary consists of the information about the space occupation [1, 4, 5] or [8].

Another view emphasizes only the final shape of the plant on a desired level of accuracy. The speed of such generation of the model is the strongest requirement in this kind of modeling. This is always in contradiction to the requirement of the quality of the model [10, 22].

The ability of simulation of the plant development [2, 5, 7, 11, 12, 15] is the next way how to classify these methods. In this paper we will point out those methods which take into account ability of interaction with environment. We will discuss several aspect of simulation on level of buds and we will focus in the size of the bud. Then we will introduce a new algorithm for the estimation of light. We will also briefly present a continuous model of simulation.

2 Previous work

In 1968 Lindenmayer introduced a definition of a string rewriting system which he used for simulation of development of multicellular organisms [9]. In 1984 Smith used this formalism for generation of the plants [19] and subsequently named this method *L-system*. The result of this simulation which is "almost" fractal he denoted *graftal*. The theory of *L-systems* has been extensively developed. The last works include continuous simulation of development [11], using differential equations and ability to interact with the environment using so called *query modules* [12, 13]. The query modules are components of the rewriting process and have the form

$$?X(x, y, z);\ \ X = P, H, U, L \tag{1}$$

where H, U, L are axis and P denotes a position of a local coordinate system of element in 3D space. Parameters x, y, z are attributes of query module. The values of the attributes are set when the process of rewriting asks the query model for them. The independence of structural and geometrical representation which is one of the advantages of *L-systems* is lost here. We must always construct the whole plant when we need to rewrite the string.

Reeves introduces *particle systems* in [16] for purposes of the film industry. He uses particles for the generation of stochastic models of grass and forests two years later [17] and does not concentrate on detail of the individual plant. He also uses special methods for the rendering of such groups of plants where the light is scaled down by exponential function in the forest [17] page 317. Although the light parameters of the scene are simulated they are not used when the plants are generated. The result of this model was again used in movies.

Very impressive results were obtained in voxel space [5]. Greene performs stochastic growth where multiple trials are attempted. Position of a growing element and its orientation are randomly perturbed, fitness function is evaluated and the best position and orientation are used as a result of the new growth direction. For every element the distance from the "center of mass" is evaluated. If the trial which lies the closest distance from the obstacle is used the effect of climbing plants is very easily obtained in this simulation. The amount of light is achieved 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 a relative number of occluded and free rays. The effect of heliotropism is achieved by constructing the illumination table at each node. Several rays are cast to cover as much from the sky as possible. The black table entries represent obstructed rays and the white unobstructed ones. The low pass filter is then used and the "hot spot" on the sky is found. This place represents the direction of further growth of the growing element.

An alternative method of the light direction estimation is represented in work [7]. Chiba et al. work on the assumption that the amount of the 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 the number of leaves in the leaf ball. The light estimation is determined by projection of these balls to the celestial sphere. The estimation is executed by applying the hidden surface algorithm or 3DDA in voxel space. In [7] the brightest direction is defined as a sum of the participating vectors associated with each ray which reaches the celestial sphere.

Holton's philosophy of *strands* comes from the fact that every branch consists of a collection of threads [6]. These threads run indivisible from root to the leaves. To describe the topological structure we need to know only the branching probabilities. About 2000 - 8000 strands are necessary to generate a realistic model. Bézier curves are used as the axis of generalized cylinders when branches are generated. The biggest advantage of this method is its clearness. We do not need very wide experience to describe the model of the plant. Holton also included in his model:

- *gravicentrism* – the tendency of the stem of the plant to grow against gravity,
- *gravimorphism* – the tendency of the branches to grow against or with the gravity,
- *phototropism* – the tendency of branches to grow in direction of the light,
- *orthotropism* – the tendency of the plant to grow vertically upward,
- *plagiotropism* – the tendency to grow in a horizontal direction,
- *planatropism* – the tendency of the plane defined by two branches to be perpendicular to the axis of the parent branch.

Although he mentioned in the case of phototropism that the light should be evaluated for every branch [6] page 61, Holton does not say how to do that.

3 The growth model

In our work we do not concentrate on precise geometry of the plant. We use lines as a branches and simplified leaves consisting of several triangles. This representation is then used as a skeleton of the model introduced by [1], which can be ray traced.

We use the stochastic model according to de Reffye at al. [2]. This model works on the level of *buds* (see figure 1). The bud is a basic growing element which can perform several actions. The action depends on external (environmental) and internal conditions of the plant. A bud can either

– die or
– bloom and die or
– sleep for a while or
– become an internode (see figure 2).

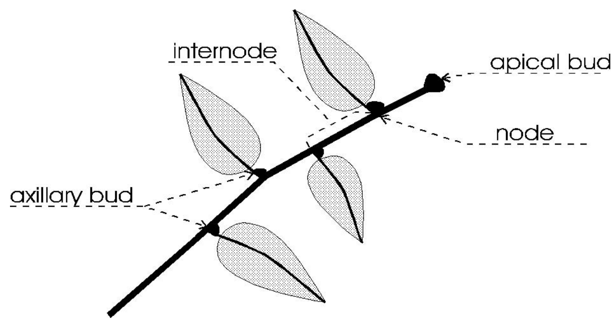

Fig. 1. A branch model according to de Reffye.

One of the problems of plant simulation is the lifetime of the plant. We have extended the model [2] in such a way that every bud has a defined *size*. The size corresponds to the amount of apical meristem allowing a bud to grow. This size consequently corresponds to the lifetime of the bud and controls the length of the branch.

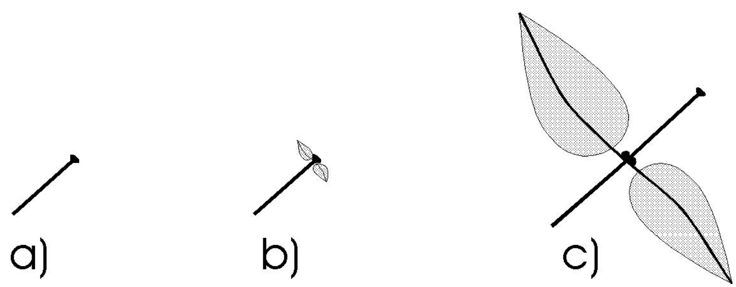

Fig. 2. Apical bud becoming an internode. a) Apex with an apical bud. b) Several leaves and lateral buds appear at their axil. c) The apical bud produces a piece of stem.

Until the bud can grow it produces axil buds with orientation depending on phylotaxis. The size of the generated buds depends on the size of the generating apical buds. The size of the new bud should always be smaller than the size of the apical bud. If the size of new bud is the same or even bigger than the size of apical bud we obtain infinite growth. Denote $a, b \in \mathcal{R}^{+} \cup \{0\}$ to be size of the apical and of the axial bud respectively. Having

$$b = \frac{a}{k}; \; k > 1 \tag{2}$$

the length of branches at different orders forms a geometrical succession.

Another phenomenon we can simulate using size of buds are different branching patterns. There are three basic types of branching patterns in nature: *basitonic, mesotonic* and *acrotonic* [13] page 19. All of these cases are shown in figure 3.

Assume linear time flow and $t \in \langle 0, 1 \rangle$ to be a lifetime of the bud a. In case of acrotonic growth an axial bud b will have size

$$b = t \; a \tag{3}$$

Changing the size of the axial bud b according to the following formula we obtain the basitonic growth:

$$b = (1 - t) \; a \tag{4}$$

And finally mesotonic growth can be described with:

$$b = \left(\frac{1}{2} - \left| \frac{1}{2} - t \right| \right) \; a \tag{5}$$

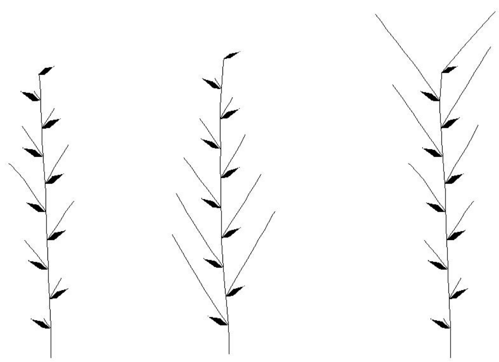

Fig. 3. Different branching pattens generated using different sizes of the bud: (from left) mesotonic, basitonic and acrotonic.

3.1 Estimation of the light

Light estimation is a very important part in the simulation of plant development. The buds will die due to lack of light coming to the leaves or become a bloom if the amount of the light is enough. The branch has a tendency to raise leaves and flowers to the light so new growth direction depends on the direction of the highest quantity of the light that is, on the position of a brightest spot as watched from the growth element.

Evaluating the amount of the light. We use sampling methods for the light estimation. The sampling direction is opposite to that in the works [5] and [7]. We sample from the sky to the fixed point (e.g the basis) of the plant. One sample consists of moving the camera to the proper point on the hemisphere and having a look at the plant. Using the Z-buffer algorithm we see from this view only those parts of the plant which are affected by the light coming from the direction corresponding to the position of the camera.

The best advantage of this method is independency of the number of samples to the number of objects in the scene. The number of samples is always constant. The quality of the sampling is affected only with the size of the sampling area (see figures 4 and 5).

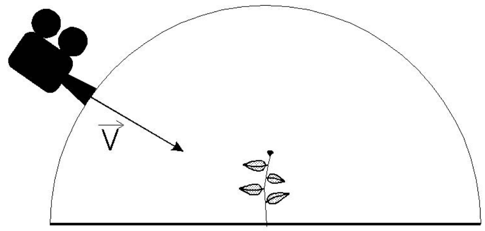

Fig. 4. Sampling of the plant from the hemisphere.

We must ensure in our algorithm that every leaf in the scene is unique. The ordering number in which a leaf was born is assigned to every leaf. During the sampling phase the plant is rendered in such a way that every leaf has assigned a different color according to its ordering number. This is easily achieved with a simple mapping function between RGB space and the ordering number. Only the leaves have different color. Every other elements have the same color as a background, so they may occlude the light.

What is essential for further computation is the area of the leaf which is visible from the sampling direction. This area is computed from the *histogram* of the sampled picture (see figure 5). The histogram is a vector of absolute frequencies of values in the picture. In our case the histogram corresponds to the visible area of the leaves projected to the plane perpendicular to the projection direction (vector v at picture 4).

We need to know a relative number of exposure of every leaf. Denote $e_i \in \langle 0, 1\rangle$ to be the relative number of exposure of leaf i. Value zero corresponds to absolute hiddenness and value e_i equal to one corresponds to total exposure of the leaf. Denote by h_i a value obtained from the histogram for the leaf i. Because *parallel projection* [3] was used we know that the maximal possible value h_i^{max} obtained for the leaf i might be

$$h_i^{max} = a_i * \frac{res_x * res_y}{ortho_x * ortho_y} \qquad (6)$$

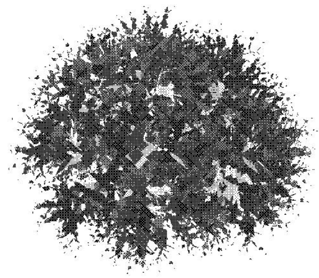

Fig. 5. A snap from the sampling camera with different color for every leaf assigned.

where a_i is the absolute area of the leaf in the space of camera, res_x, res_y are values of the resolution of the sampled snap in pixels and $orho_x, ortho_y$ are values of the camera viewport.

There are several tips on how to evaluate the area of the leaf a_i. A very straightforward way is to approximate the leaf by several triangles and compute the area of every triangle in the following way. Let $t_j; j = 1, \ldots, n$ be a triangular approximation of the leaf. Every triangle t_j consist of points x_j, y_j and z_j. The area s_j of the triangle can be calculated using the vector product:

$$s_j = \frac{|(x_j - y_j) \times (y_j - z_j)|}{2} \tag{7}$$

The area of the leaf i will be:

$$a_i = \sum_{j=1}^{n} s_j . \tag{8}$$

The relative number of exposure can be easily evaluated from equation (6) and the histogram value h_i;

$$e_i = \frac{h_i}{h_i^{max}} \tag{9}$$

The total amount of light coming to the leaf would be increased by the amount of ambient light presented in the crown of the tree. In our simulation we use the value 0.1.

The only problem is with aliasing. If the leaf becomes smaller than one pixel it is invisible on the display. This situation is represented in the histogram with a value of zero and the leaf should be removed. We have stated a so called "preserving time" for every leaf. This means if the area of the leaf is less than some constant we state that it has total exposure.

Suppose an artificial example with one direction of light coming from the top and no ambient light present. Assume the leaf will die due to lack of light if there is less than 5% exposure. Figure 6 shows the shedding due to lack of light.

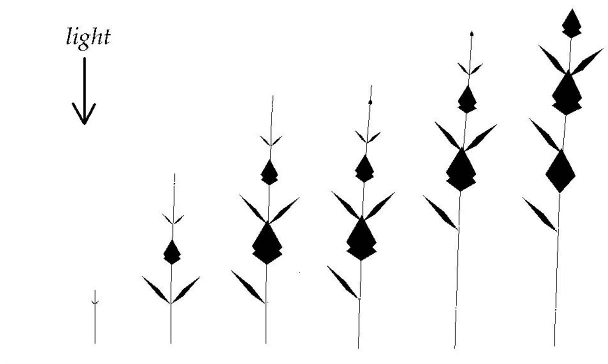

Fig. 6. Process of dying of leaves due to lack of the light.

Evaluating the growth direction. Chiba et al. state [7] on page 6:

> ...the brightest direction vector is defined as *the sum* of the vectors each of which has the same direction as each ray which reaches the celestial sphere...

This approach leads to wrong results as can be seen in the 2D example in figure 7. Having one leading shoot and an obscuring plane over it we obtain two brightest

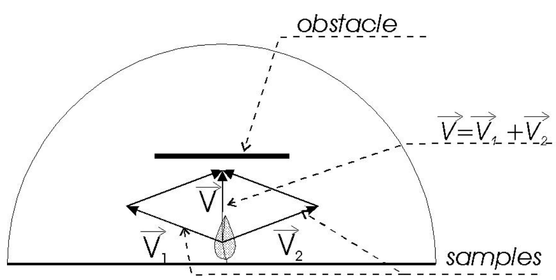

Fig. 7. Estimation of the brightest spot on the sky using sum of the samples leads to unadequate results.

direction vectors v_1 and v_2. Their sum v is direct to the center of the black space.

In our simulation the direction vector for every sample has an opposite sign to the potential growth direction (see figure 4). The brightest direction vector is the vector from the direction which brings the maximal amount of light from all of these samples. If there is more than one possibility one of them is *randomly* chosen. In the case of our example from the figure 7 we would choose one of the vectors v_1 or v_2.

The direction vector of the leaf with maximum exposure nearest to the bud is used as a new growth direction. We modify this direction in such a way as in [6], it is: the new direction vector v_{new} is computed as a vector sum of the old one and of one giving the direction of the brightest spot s multiplied by the coefficient of sun seeking k_s:

$$v_{new} = v_{old} + k_s s. \tag{10}$$

Figure 8 demonstrates a change in the shape of the plant under the very strong light source coming from the direction $(1, 1, 0)$.

Fig. 8. Left figure demonstrates plant with phylotaxis $2\pi/3$ and acrotonic growth. On the right is the same plant under the influence of very strong light coming from the direction $(1, 1, 0)$

There are several ways to sample the sky. Greene [5] samples the sun trajectory and Chiba et al. [7] state that the sky has constant exposure at its every point. We must estimate the relative weights of the samples depending on the duration of the simulation. These weights represent the percentage of the maximal possible amount of the light coming from one direction. It is obvious that

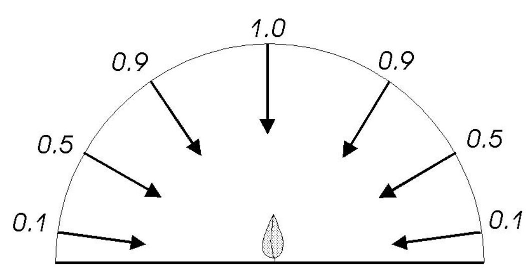

Fig. 9. The relative weights of the samples are smaller in the morning than at noon.

for a long period there is no reason why to sample the sun trajectory. But on the other hand, the power of the sun is less in the morning than at noon. We propose a solution as shown in figure 9.

Figure 10 demonstrates the following two approaches. A constant amount of light from every side was used on the right picture. The left picture demonstrates the sample weighting according to figure 9. This result seems to be more realistic.

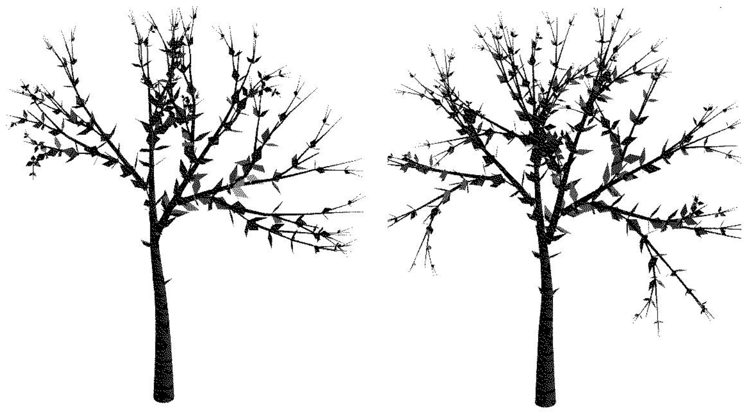

Fig. 10. Left figure demonstrates plant with exposure weighting, on the second one constant light coming from all of the sides was used

4 Simulation algorithm

We use continuous simulation in our model. Input to the algorithm is a desired number of frames, duration of the simulation and parameters of the plant. We use an algorithm with a discrete time step. This method works on the assumption that in every time slice $\langle t, t+\delta)$ the environment conditions do not change. Events like ramification and death of the bud, the leaf or the branch are inherently discrete. They can occur anytime in the time slice. We use a method called "event planning" for discrete events. The idea of this method is following: we find first discrete event which will occur in the actual time slice. This event is performed and new first event is found. This cycle ends at the end of the time slice. We have done all of the discrete events and we must perform all of the continuous events (the growth of stems and leaves) at the time $t+\delta$.

The algorithm scratch has the following form:

- compute the duration of one time slice:
  $\delta = lengthOfSimulation/numberOfFrames$
- for every time slice $\langle t, t+\delta)$ do
  - estimate the amount of light for every leaf.

  - do every discrete event in $\langle t, t+\delta)$
  - growth continuous elements to time $t+\delta$
  - generate geometry
  - save script for ray tracer
- increase time by δ

5 A note about artificial life

Smith [19] mentioned the term *database amplification* also cited by [3] and [13]. Database amplification is the possibility of generating complex structures from small data sets. The second study [13] also state that this term is related to *emergence* what is one of the central themes of the study of the artificial life [20]. Emergence is a process in which a collection of interacting units acquires qualitatively new properties [20].

We can look to the plant modeling as described in this paper also from this viewpoint. The buds are the interacting entities. They compete together for the light obtaining a positive fitness value (see also [18]) if the light is reached. The worst are removed from the simulation. The final shape of the plant is the emergent phenomenon.

6 Implementation and results

Silicon Graphics Indigo^2 with Extreme graphics board and R4400/200MHz has been used. We use hardware *Z-buffer* for the light estimation. A generating of a sequence consisting of 200 frames of the plant with the final number of leaves about 3000 and 50 samples per frame takes approximately two hours. The bottleneck of this method is the histogram evaluation. On the other hand even only 10 samples bring valuable results and the difference between 50 and 500 samples is below the level of recognition. A sampling area on 400×400 pixels resolution was used.

7 Conclusion

A new method for estimation of the amount of light and growth direction of the growing elements of the plant has been presented in this paper. This method uses taking snaps of the plant from the celestial hemisphere and evaluating the histogram for calculation of the exposure. The number of samples is independent to the number of objects in the scene. We have also presented the size of the bud as a control mechanism for the generating of the branching pattern of the plant.

8 Acknowledgements

We would like to thank Mr. Jiří Žára, Mr. Pavel Slavík, Mr. Aleš Holeček and Mr. Jan Přikryl for their valuable discussion.

References

1. J. Bloomenthal. Modeling the mighty maple. In *Proceedings of SIGGRAPH '85, in Computer Graphics Proceedings, Annual Conference Series 1985*, volume 19(3), pages 305–311, 1985.
2. 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, in Computer Graphics, Annual Conference Series 1988*, volume 22(4), pages 151–158, 1988.
3. J. Foley, A. van Dam, and J. Hughes. *Computer Graphics: Principles and practice.* Addison-Wesley, Reading, 1990.
4. D. Fowler, P. Prusinkiewicz, and J. Batjes. A collision-based model of spiral phyllotaxis. *ACM Computer Graphics*, 26:361–368, 1992.
5. N. Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. In *Proceedings of SIGGRAPH '89, in Computer Graphics Proceedings, Annual Conference Series 1989*, volume 23(4), pages 175–184, 1989.
6. M. Holton. Strands, gravity and botanical tree imagery. *Computer Graphics Forum*, 13(I):57–67, 1994.
7. 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.
8. Y. Kawaguchi. A morphological study of the form of nature. *ACM Computer Graphics*, 16(3):223–232, 1982.
9. A. Lindenmayer. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, Parts I and II(18):280–315, 1968.
10. P. Oppenheimer. Real time design and animation of fractal plants and trees. In *Proceedings of SIGGRAPH '86, in Computer Graphics, Annual Conference Series 1986*, volume 20(4), pages 55–64, 1986.
11. P. Prusinkiewicz, S. Hammel, and E. Mjolsness. Animation of plant development. In *Proceedings of SIGGRAPH '93, in Computer Graphics, Annual Conference Series 1993*, volume I, pages 351–360, 1993.
12. P. Prusinkiewicz, M. James, and R. Měch. Synthetic topiary. In *Proceedings of SIGGRAPH '94, in Computer Graphics, Annual Conference Series 1994*, volume I, pages 351–358, 1993.
13. P. Prusinkiewicz, M. James, R. Měch, and J. Hannan. The artifical life of plants. In *Course Notes of SIGGRAPH '95, in Computer Graphics, Annual Conference Series 1995*, volume I, pages 1–38, 1995.
14. P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants.* Springer-Verlag, New York, 1990.
15. P. Prusinkiewicz, A. Lindenmayer, and J. Hanan. Developmental models of herbaceous plants for computer graphics imagery purposes. In *Proceedings of SIGGRAPH '88, in Computer Graphics, Annual Conference Series 1988*, volume 22(4), pages 141–150, 1988.
16. W. Reeves. Particle systems – a technique for modeling a class of fuzzy objects. *ACM Transaction on Graphics*, 2(2):12–22, 1983.
17. W. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *Proceedings of SIGGRAPH '85, in Computer Graphics, Annual Conference Series 1985*, volume 19(3), pages 313–322, 1985.

18. K. Sims. Evolving 3D morphology and behavior by competition. In *Artificial Life IV Proceedings*, volume I, pages 28–39, 1994.
19. A. Smith. Plants, fractals and formal languages. In *Proceedings of SIGGRAPH '84, in Computer Graphics, Annual Conference Series 1984*, volume 18(3), pages 1–10, 1984.
20. C. Taylor. *"Fleshing out" Artificial Life II*, volume II. Addison-Wesley, Redwood City, 1992.
21. X. Viennot, G. Eyrolles, N. Janey, and D. Arques. Combinatorial analysis of ramified patterns and computer imagery trees. In *Proceedings of SIGGRAPH '89, in Computer Graphics, Annual Conference Series 1989*, volume 23(3), pages 31–40, 1989.
22. J. Weber and J. Penn. Creation and rendering of realistic trees. In *Proceedings of SIGGRAPH '95, in Computer Graphics Proceedings, Annual Conference Series 1995*, volume 22(4). ACM SIGGRAPH New York, 1995.