Fast Algorithm for Stochastic Tree Computation

MengZhen KANG[1,3] Philippe de REFFYE[3,4] Jean François BARCZI[2,3] BaoGang HU[1,3]

1 NLPR, Institute of Automation, CAS, 2728# Box, 100081, Beijing, China

2 CIRAD, Monpellier, France

3 LIAMA, Institute of Automation, CAS, 2728# Box, 100081, Beijing, China

4 INRIA, Paris, France

ABSTRACT

In this paper, a recursive algorithm that can build complex tree structure quickly is presented briefly. The plant structural growth is based on a dual-scale automaton, in which 'macrostate' and 'microstate' are used to simulate the growth unit and the metamers inside growth unit separately. Each state is characterized by its 'physiological age'. The automaton is non-deterministic in case that the probabilities exist in each growth cycle simulating the bud activity. Since the main plant structure and branches are statistically similar, and each structure can be decomposed into an axis and substructures, simulation is begun from smallest structures that have no further branch, and that information are used when such kind of structures appear in other branches. To keep the diversity of tree structure, several stochastic structures will be simulated at one time for a given physiological and chronological age. This algorithm can save a lot of time compared to node-by-node simulation. Verification is done both in organ number counting and visualization.

Keywords

Stochastic, substructure, tree growth, 3D plant simulation, fast algorithm

1. INTRODUCTION

Since plant structures are diverse and irregular, stochastic geometrical parameters are often used in generating naturally looking image. For example, Reeves[Reev85] used random geometrical parameters drawn from uniform distribution to create stochastic trees with particle systems. Oppenheimer[Oppe86] drew statistically self-similar tree by setting mean and standard deviation for parameters.

Another kind of stochastic plant model was initiated by de Reffye[Reff88]. Knowledge of bud activities was considered to simulate stochastic growth of plant. These concepts are used in AMAPSim software [Reff90], and also recently in GreenLab model in which a new dual-scale automaton was developed [Zhao01] to create plant topological structure.

To generate large forest or big trees takes long CPU time. Instancing was used for representing complex scenes to enhance the speed of creation. The basic idea of instancing is to represent similar objects with same data so that data storage drops down while the reduction of diversity cause only negligible visual impact. Instancing can be used in different levels thanks to hierarchical structure of plant and forest scene. This method was used in [Deus98] [Hart92] [Reev85] to create plant figure. Recently, Yan [Yan02] presented an algorithm to construct rigid tree structure based on substructures.

The current paper is a progress in the frame of GreenLab model. The automaton that is originally defined in [Zhao01] is used here as the topological model. In this context, a new way of instancing—stochastic substructure—is developed to create quickly diverse tree structure. The number of organs in each substructure can be calculated, as well as their theoretical mean and variance. The result shows that this algorithm is much quicker and keeps same visual effect compared to node-by-node simulation.

2. STOCHASTIC SUBSTRUCTURES

2.1 Growth Probabilities

2.1.1 Chronological and physiological age

Each step of growth (a *growth cycle*) corresponds to a *chronological age*. It can be one year or one month for real plant. Another important notion is

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972

WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

physiological age, which is used to distinguish different states of growth in plant. For example, a branch grows after its main axis has appeared, so its physiological age is bigger than the axis, while its chronological age is smaller. In a complex tree, the maximum physiological age may be 5 or 6.

2.1.2 The dual-scale automaton

The minimum unit of plant that we simulate is a *metamer*, which is composed by an internode and its axillary leaves, fruits and buds. Metamers that appear on the top of an axis in same growth cycle make up a *growth unit*. The automaton simulates both kinds of unit by *microstate* and *macrostate* at two scales. Each macrostate is done by an automaton running on microstates belonging to it, as illustrated by the bigger ellipse in Fig. 1. Each microstate inside the smaller ellipse is composed by an internode and its axillary buds that can have different physiological age, which can decide what kind branches the buds will become. The numbers above each state shows how long that state will keep in same state before transition, and the arrow under them shows which state it will jump to. The initial state is empty, and each automaton run from left to right.

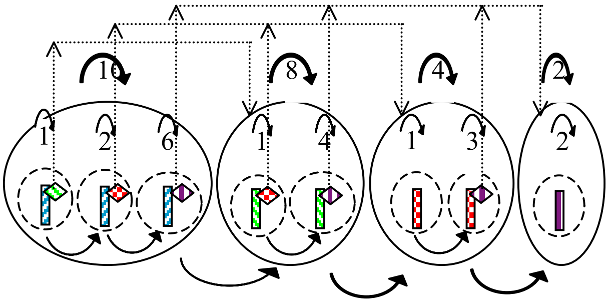


Figure 1 An example of automaton

2.1.3 Probabilistic growth

At a given cycle, a bud can die, or rest, or produce growth unit which different number of metamers. We can simulate these phenomena by adding probabilities for each arrow in Fig. 1. The run of macrostate can continue according to *survival probability* p_c, and if it still can run, it create a growth unit by *growth probability of macrostate* p_b. The metamers inside it appear by *growth probability of microstate* p_u. The probabilities can be different according to the physiological age. Since if a bud die, there is no further unit on that axis, the survival probability has strongest effect on final shape.

2.2 Simulation with Stochastic Substructures

Substructure is a recursive concept. Each structure is composed of an axis and branches that are called substructures. The substructures are characterized by their physiological and chronological age. The simulation begins from the maximum physiological age of plant p_m to 1, which is inversed to the automaton run sequence. For each physiological age, structures with chronological age changing from 1 to N (N is age of plant) is simulated. The last structure done is then the tree, with physiological age 1 at chronological age N.

In context of stochastic growth, with given chronological age and physiological age, there are numerous different structures. However, human eyes can't see the difference if there are several same branches in a complex tree. So for a given chronological age and physiological age, a limited set of different structures are created. Each axis are simulated by Monte-Carlo method considering probabilities $p_{b,}$ p_c and p_u, while the branches are chosen from corresponding substructure sets. The number of substructures in each set, or *set size*, can be different with physiological age, and of course, the smaller the set is, the faster the simulation will be.

Fig. 2. illustrates some stages in the procedure of the simulation. The set size for substructures T is 5, and the maximum physiological p_m is 4. The simulation began with physiological age 4, which has no branches. *S(i,j)* means substructure with physiological age i at chronological age j. For example, in constructing set of substructure *S(3,3)*, on the first growth unit from the bottom, substructure *S(4,2)* will be needed as a branch, and each of them is chosen randomly from the five substructures in set *S(4,2)*.

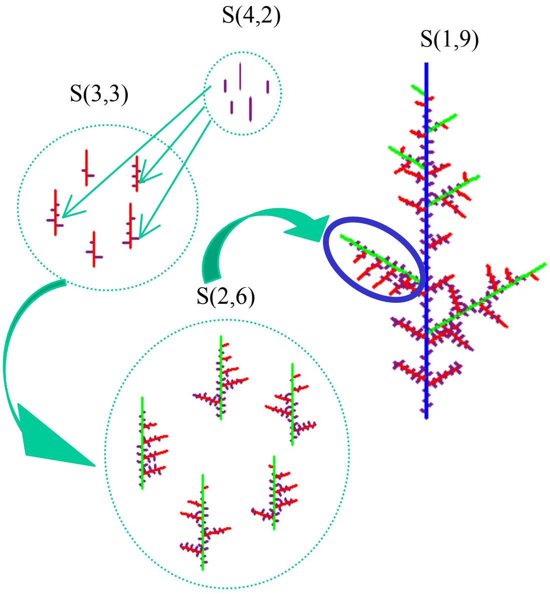


Figure 2 Simulation procedure

3. RESULTS

3.1 Number of Produced Metamers

Metamer is the minimum unit in this model, and number of organs like leaves can be got if number of metamer is known. The distribution of this number is a compound law because of the probabilities. Luckily we have deduced the formula that calculates the mean and variance independent with the simulation procedure. The tedious demonstration that gives these results is not to be described here. We simply give results to show that the simulation based on stochastic substructures is reasonable.

Practically speaking, a small set size is desirable and sufficient to get high performance and good visual effect. Let the set size of substructures with physiological ages greater than one be 5 here. We give an example based on a similar automaton like in Fig. 1. The probabilities are combination of those given, that is, $p_b=0.8$, $p_c=0.9$, $p_u=0.8$ *and* $p_a=0.5$. The plant age is 24. Total 100 trees are simulated.

Age_Ph	M_S	V_S	M_Th	V_Th
4	69.7	3109.7	58.5	2362.6
3	49.7	1812.9	47.7	1714.6
2	36.5	1240.0	35.3	1237.5
1	33.4	524.8	33.7	528.2
Total	189.3	22431.5	175.3	20174.6

Table 1. Simulation vs. theoretical results

In Table 1, *Age_Ph* is the physiological age of metamers, *M_S* and *V_S* are the statistical mean and variance of simulation, while *M_Th* and *V_Th* are the corresponding theoretical value. Despite the small set size, there is little shift between the two results. This means simulation with stochastic structure is good approximation. With set size increase, the simulation result converges to the theoretical value.

3.2 Simulation Performance

Our software runs on Matlab, on a PC with an Intel Pentium 4 processor at 1700MHz, system memory 256MB. The performance is checked on the topological calculation.

We set all the probabilities to one to make sure each bud will be visited. The set size for each substructure is still 5 and 100 trees are simulated. We compare it with node-by-node simulation, in which trees are simulated from bottom to top and each node will be simulated.

We have known that the substructure sets should be constructed before the first main tree is done. This procedure takes some time. But then the substructure sets can be reused repeatedly for all other trees. So in average, time spent for each tree is just for simulation of its axis and picking substructures from sets. Compared to node-by-node simulation, time spent for one tree is already short, and the average time for a tree in a stand is even less.

Table 2 lists the performance comparison of average time spent for each of 100 trees. We can see that the performance ratio is more than 700 at age 30. For 100 trees at age 30, time spent with substructure algorithm is less than 45 second, but with prefixed order, CPU time is 27,747 seconds, which is enormous and unbearable.

N	Sub	Pre	Nb_O	Ratio
5	0.010	0.20	2,440	20.0
10	0.041	3.14	4,480	76.6
15	0.093	17.69	238,120	190.2
20	0.173	58.37	775,360	337.3
25	0.279	132.14	1,928,200	473.6
30	0.387	277.47	4,048,640	716.9

Table 2. Average CPU time for the a tree

3.3 Visual Results

Not only topological computation, but also 3D structure of plant can be done with substructures. The 3D substructures are reused after Euclid transformation. Here we simply give the result without the detail of geometrical parameters and operations.

One result given is shown in Fig. 3. It is from an automaton similar with the one defined in Fig. 1. Only internodes are displayed.

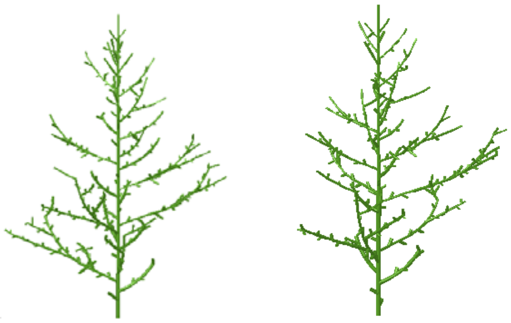
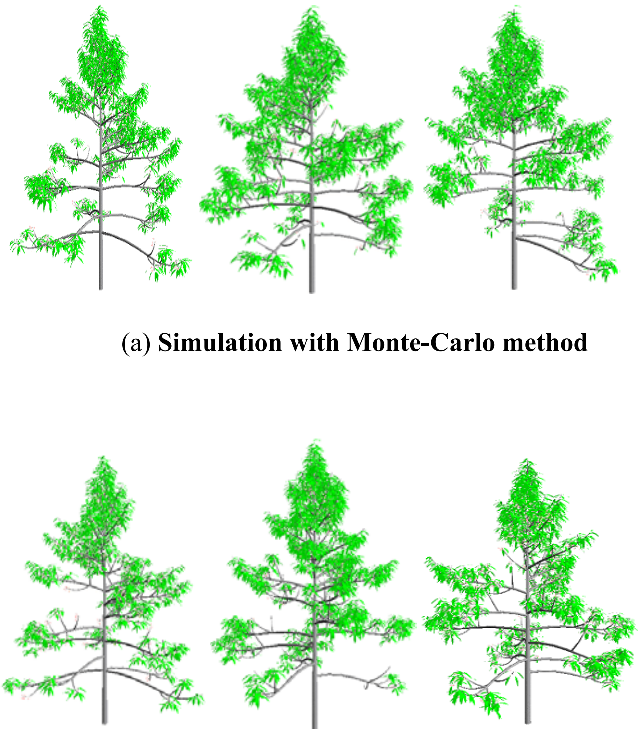


Figure 3 Stochastic tree structures from simulation

Another result is shown in Fig. 4. It is from AMAPSim(written in C language) software, to compare classical stochastic trees built from a pure Monte-Carlo method with crude stochastic trees generated by a limited set of substructures. The AMAPsim software can simulate stochastic trees with 2 repetitions for each substructure of same kind. The Monte-Carlo method generates a single 20 years old tree in 2 minutes, while the substructure method needs only less than one second. So the gain for this tree is about 200.

AMAPSim is based on reference axis, not the automaton we introduced before. It means that such algorithm may be used for other models, for example, generating tree with character rewriting system.



(a) **Simulation with Monte-Carlo method**

(b) **Simulation with Substructures**

Figure 4 Cherry tree 20 years old done by AMAPSim

So in display, the small set size can bring nice visual effect. It is even difficult to distinguish two results from different set size. This indicates a great advantage of substructures over the classical way. It allows simulating a forest stand of wild Cherry trees in a reasonable amount of time.

4. DISCUSSION

Stochastic substructure is a kind of instancing, but the instanciation is done by choosing a substructure from set randomly at topological level. It can break the bottle neck remained for computation of big trees, root systems or even a forest. The larger the forest the more efficient the algorithm is. But of course this algorithm has no obvious advantage for plants with simple architecture like maize or sunflower.

The advantage of this new method lies also in that it lay more on mathematical calculation than simulation itself. The theoretical mean and variance of the tree production is a great advantage because it gives a mathematical background that frames the simulation.

Applications of this model can also be for agronomy, for example, to simulate organs production in a plantation, or to provide good calculation for the light interception into a canopy.

5. ACKNOWLEDGMENTS

This research is supported by LIAMA (Sino-French Laboratory in Information, Automation and Applied Mathematics), CASIA (Institute of Automation, Chinese Academy of Sciences), and Natural Science Foundation of China (60073007).

6. REFERENCES

[Deus98] Deussen, O. Hanrahan, P. Lintermann, B. Mech, R. Pharr, M. Prusinkiewicz, P. Realistic modeling and rendering of plant ecosystems. Computer Graphics 32, No. 3, pp. 275-286, 1998

[Hart92] Hart, J. C. The Object Instancing Paradigm for Linear Fractal Modeling. In proc. of graphics interface 92, pp. 224-231, 1992

[Oppe86] Oppenheimer, P. E. Real time design and animation of fractal plants and trees. Computer Graphics 20, No.4, pp. 55-64, 1986

[Reev85] Reeves, W. T. and Blau, R. Approximate and probabilistic algorithms for shading and rendering structured particle systems. Computer Graphics 19, No. 3, pp. 313-322, 1985.

[Reff88] de Reffye, Ph. Edelin, C. Francon, J. Jaeger, M. Puech, C. Plant models faithful to botanical structure and development. Computer Graphics 22, No. 4, pp.151-158, 1988.

[Reff90] de Reffye, Ph. Dinouard, P. Jaeger, M. Basic concepts of computer plant growth simulation. In: Computer Graphics: "Where do we go now that we've arrived? NICOGRAPH' 90, Tokyo (JPN), Nov., pp. 219-234, 1990

[Smit84] Smith, A R. Plants, fractals and formal languages. Computer Graphics 18, No. 3, pp. 1-10, 1984

[Sole02] Soler, C. Sillion, F. Blaise, F. de Reffye, Ph. An efficient instanciation algorithm for simulating radiant energy transfer in plant model. ACM transaction on graphics, 2002, in press

[Yan02] Yan, H.P. Barczi, J. F. de Reffye Ph. and Hu, B.G. Fast Algorithms of Plant Computation Based on Substructure Instances. In WSCG' 03 conf.proc. 3, No. 10, pp.145-153, 2002.

[Zhao01] Zhao, X. de Reffye, Ph. Xiong, F.L. Hu, B.G. Zhan, Z. G. Dual-scale automaton model for virtual plant development. The Chinese Journal of Computers 24, No. 60, pp.608-617, 2001. (Chinese in English abstract)