HAL
open science

Stochastic 3D Tree Simulation Using Substructure Instancing

M. Z. Kang, Philippe de Reffye, Jean-François Barczi, B.G. Hu, François Houllier

To cite this version:

M. Z. Kang, Philippe de Reffye, Jean-François Barczi, B.G. Hu, François Houllier. Stochastic 3D Tree Simulation Using Substructure Instancing. Plant Growth Modeling and Applications, LIAMA: The Sino-French joint Laboratory In Computer Science, Control And Applied Mathematics / CAU :ChinaAgricultural University, Oct 2003, BeiJing / China, China. pp.154-168. inria-00121486

HAL Id: inria-00121486

https://inria.hal.science/inria-00121486v1

Submitted on 12 Jan 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stochastic 3D Tree Simulation Using Substructure Instancing

M.-Z. Kang[1,3,6], P. De Reffye[2,3,4], J.-F. Barczi[2,3], B.-G. Hu[1,3], F. Houllier[2,5]

[1] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, BP2728# Box, 100080, Beijing, China*

[2] *UMR AMAP, CIRAD, TA 40/PS2, 34398 Montpellier Cedex 5, France*

[3] *Sino-French Laboratory in Information Automation and Applied Mathematics, Institute of Automation, Chinese Academy of Sciences, BP2728# Box, 100080, Beijing, China*

[4] *INRIA, Projet Metalau, Rocquencourt, France*

[5] *INRA, Département Forêts et Milieux Naturels, France*

[6] *Information Engineering College, Capital Normal University, 100037, Beijing, China*

Abstract

Tree growth is simulated using a stochastic model of organogenesis that is faithful to botanical knowledge. This model is based on the concept of bud "physiological age", on the statistical description of the transition from one physiological age to another as well as of bud death, bud growth and branching processes. In order to enhance simulation efficiency, a recurrent algorithm based on stochastic substructure instancing is proposed. The tree is hierarchically decomposed into substructures that are classified according to their physiological age, and a library of random substructure instances is constructed: the recurrent simulation starts with the simplest peripheral substructures, which are also the physiologically oldest; these substructures are then progressively assembled into more complex substructures, until the tree is completely simulated. When the size of the library of substructure instances is small, the time needed to build a single stochastic tree is much shorter than for a usual tree simulation that operates on a bud-by-bud basis. in computing a group of trees, the speed gain is even much greater, because the library of substructure instances is built for the first tree, and then is reused for computing subsequent trees. A preliminary sensitivity analysis is carried out according to the size of the library: when the library is large, the simulated distribution of the number of organs fits well with the theoretical mean and variance; the algorithm can thus be tuned in order to obtain accurate predictions. On the other hand, a small library (e.g., with only 2 or 3 instances for each substructure class) is sufficient for generating visually realistic trees. A few examples illustrate the high performance of this algorithm which paves the way for the fast simulation of large forest scenes.

Keywords: stochastic model, simulation, instancing, substructure, tree architecture, 3D virtual plant, fast algorithm

1 Introduction

In recent decades, a lot of algorithms have been developed for simulating natural phenomena. As for plants and landscapes, various methods have been elaborated for generating realistic images and animations; these methods have been applied, for example, in the field of human entertainment (e.g. computer games and films)[1], as well as in agronomy[2]. Since trees are diverse and irregular, stochastic geometrical parameters were used to generate "realistic" images. For example: Reeves[3] used structured particle systems for the serial computation of stochastic trees in a forest scene, in which the random geometrical parameters of each tree were drawn from a uniform distribution; in order to construct fractal plants, Oppenheimer[4] drew statistically self-similar trees by setting the mean and standard deviation of the geometrical parameters. Viennot[5] used "ramification matrix" based on hydraulic concepts to generate stochastic tree image capable of direct control on final form, but it aimed much more on visual effects than on botanical concepts.

Another type of stochastic plant models was initiated by de Reffye et al.[6]: their models are based on botanical concepts that account for plant structure and development, especially on the probabilistic analysis of bud activity[7]. These concepts were first used to design "AMAPsim", a generic plant simulation software[8, 9]; they further formed the basis of a new model of plant morphogenesis, dual-scale automaton[13], on which a new functional-structural model "GreenLab" is founded. One of the computational issue with such models lies in the generation of large landscapes, which contain a high number of plants: such simulations are usually fairly slow and require a huge amount of memory, so that there is a strong need for enhancing the performance of scene creation, i.e. for improving the trade-off between the realism of the outputs, the memory requirement, and the speed of the algorithm.

Instancing has already been used for representing complex and large plant scenes. The basic idea of instancing is (i) to identify classes of similar objects, (ii) to build a library of instances for each class of objects, and (iii) to use this library for assembling new instances in order to generate the scene. Such algorithms can result both in a decrease in memory requirement for data storage, and in a reduction of natural tree diversity whose visual impact is usually negligible. For vegetal landscapes, instancing can be used at different levels thanks to the hierarchical structure of plants. Indeed, instancing was used by various authors, who used L-systems [11], fractal models [12] and particle systems [3] for simulating plants. Recently, Yan et al. [12] adapted a deterministic version of de Reffye's models [6] and proposed a fast algorithm to compute tree topology.

The aim of this paper is to extend Yan's deterministic approach [12] to stochastic substructures, using the type of dual-scale automaton that was originally defined by Zhao [13]. This automaton is based on the hierarchical decomposition of the plant into nested substructures, which are made up of branched and/or unbranched axis, each axis being defined as the succession of growth units (or shoots), which are themselves composed of a succession of internodes. Growth units are considered as the "macro level", whereas internodes are considered as the "micro level"; the hierarchy between the substructures is ensured by physiological age. In this context, a new way of using instancing is proposed: a library of random substructure instances, which exhibit similar characters and bud functioning probabilities, is created; when simulating a tree, these instances are sampled from the library randomly and assembled at plant level. The number of organs in each

substructure instance can be calculated and compared to its theoretical mean and variance, which can be derived from bud functioning probabilities. We show that this method can dramatically speed up the generation of plant topology, as compared to serial bud-by-bud simulation of each tree, without altering the visual realism of the simulated plants.

2 Stochastic Substructures

2.1 Bud functioning probabilities

2.1.1 Definitions

Each step of growth, also named "growth cycle", corresponds to a given "chronological age". The time step corresponds to the generation of a new shoot; for real plants, it can be one year or less. The "physiological age" is another important concept, which is used to distinguish the state of the organs (buds, internodes, shoots, substructures, etc.) within the plant. For example, the physiological age of the branches can vary according to their order: high-order branches are usually older than low-order branches; similarly, within a shoot, the lateral short branches are often physiologically older than the long ones. However, branches of same physiological age can appear at different branching order. The highest order branches are supposed to have the maximum physiological age, which is 5 or 6 for a complex tree.

A shoot is composed of a series of metamers, a metamer being defined as an internode and the lateral organs which it bears (leaves, fruits, lateral buds). Within a growth unit, the internodes are supposed to appear one after another: a bud can either set in place a new bud, or rest, or die; as a consequence some potential metamers may not appear. Lateral buds can either develop into a branch or rest. These processes can be described with probabilities: plant growth is thus viewed as a stochastic process (Fig.1). The survival probability of a bud from one growth cycle to the next is noted p_c: at each cycle, a bud can indeed die because of an accident; if it dies, the growth of the axis stops. If the bud survives it can either generate a new shoot with a probability noted p_b, or rest until next growth cycle. If the bud rests, the shoot does not appear, but the bud is still living and can further grow during the next cycle (Fig.1a & Fig.1b). Within a growth unit metamers can successively appear with a probability p_u: since some potential metamers don't appear, the length of the growth units may vary (i.e., they exhibit a different number of metamers; see Fig.1c). Each metamer can also bear a certain number of lateral buds, but only some of them may develop into branches (Fig.1d): the probability for a lateral bud to develop is noted p_a. The GreenLab model is therefore defined by the structure of the dual-scale automaton (see Fig.4a for an example) and by the probabilities p_c, p_b, p_u and p_a, which usually vary according to physiological age.

2.1.2 Effect of bud functioning probabilities on plant growth

In order to simply illustrate how these probabilities can influence tree topology, we suppose in this section that they are constant for any physiological age. Fig.2e shows the structure of a tree when all probabilities are equal to 1 (i.e., in a deterministic situation). In that case, the result is the same as in Yan et al. [12]. Then, we check the effect of each probability on tree structure by setting other probabilities to ones. In Fig.2a, the growth probability of each bud, p_b, is 0.8: the number of growth units in each axis thus follows a binomial distribution. In Fig.2b, the survival probability of each bud, p_c, is set to 0.9: in that case, the number of actual growth units in an axis follows a geometrical distribution; this

example illustrates that even with a high value of this probability, the topology of the tree can vary strongly.

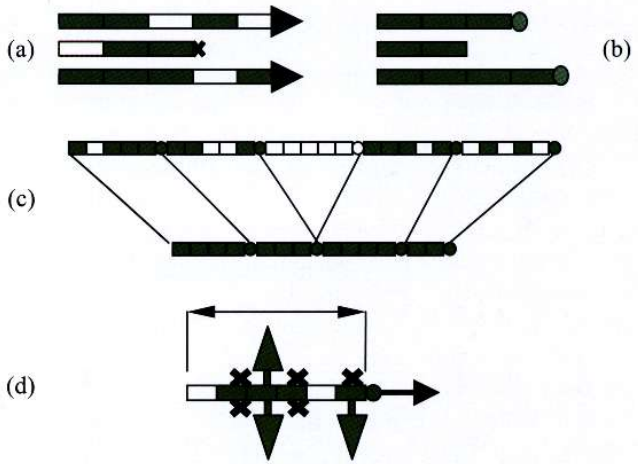
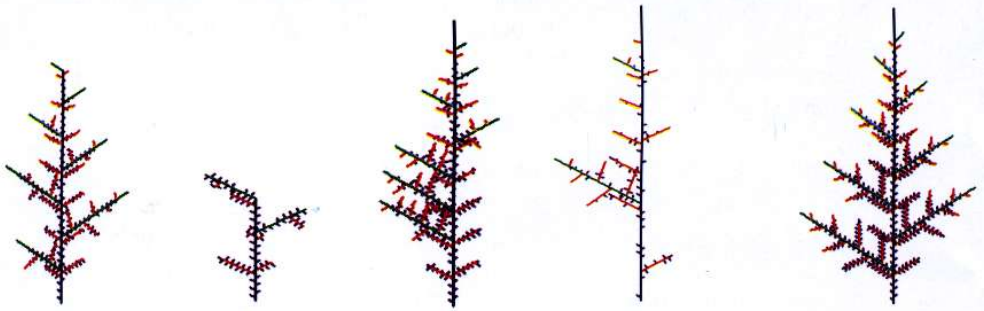


Fig.1 Plant growth: probabilities of bud functioning. (a) Virtual axis: growth of an axis over 5 growth cycles. The axis is made of a succession of annual shoots. Cross: bud death (the axis cannot grow further) with a probability $1-p_c$ at each growth cycle. Arrow: the bud is alive and the axis can continue to grow. Green segments: growth units (or shoots) that really developed, with a probability p_b. White segments: virtual growth units which did not grow because of bud dormancy, with a probability $1-p_b$. (b) Real axis, as it can be observed as the result of the growth process described in (a).(c) Detailed structure of an axis that grew over 5 growth cycles: each shoot is made of metamers. Green segments: metamers that really developed, with a probability p_u. Blank segments among green ones: virtual metamers which did not develop, with a probability $1-p_u$. (d) Branching inside a growth unit. Green arrow: the lateral bud survived, with a probability p_a, and created a substructure. Cross: the lateral bud died and did not grow, with a probability $1-p_a$

In Fig.2c, the metamers appear (within a growth unit) with a probability p_u equal to 0.8: this induces a variability in the length of the growth units. In Fig.2d, each branch lateral appears with a probability p_a equal to 0.5: as a result, the crown of the tree is less dense.



(a) p_b=0.8 (b) p_c=0.9 (c) p_u=0.8 (d) p_a=0.5 (e) p_b=p_c=p_u=p_a=1

Fig.2 Influence of bud functioning probabilities on plant pattern

Real plants result from the combination of all these probabilities and from their variation

according to bud physiological age as well as to the environmental conditions which may vary according to chronological age.

2.2 Simulation with stochastic substructures

Plant substructure is a recurrent concept: each substructure is composed of a main axis and branches (born by the main axis), which are themselves substructures. Let us consider a plant where the maximum physiological age is noted P_m and whose structure is simulated at chronological age N. This plant is made up of numerous substructures which are noted $S_{p,r}(n)$, where p is the physiological age ($1 \leq p \leq P_m$), r is an index that design the rank of the substructure repetition in the library, and n is the chronological age of the substructure ($1 \leq n \leq \min(N, L_p)$, where L_p is the maximum life span of the substructures of physiological age p).

The simulation is based on two complementary principles. (i) The plant is recurrently built as an assemblage of substructures: the simulation starts with the simplest peripheral substructures, which are also the physiologically oldest ($p=P_m$); these substructures are then progressively assembled into more complex substructures, until the complete tree is simulated ($p=1$). (ii) In order to enhance the speed of the procedure, a 'library' of substructure instances is built. This library is a subset of all potential substructures, it covers all possible physiological and chronological ages. The library is noted $\{S_{p,r}(n): 1 \leq p \leq P_m, 1 \leq r \leq T_p, 1 \leq n \leq \min(N, L_p)\}$, where T_p is the size of the subset of the substructures of physiological age p: T_p is defined by the modeler according to his performance requirements and it can vary according to p (see below). The substructures that belong to this subset are named 'instances': they are built iteratively using the already built, and physiologically older, instances which are randomly sampled from the library and then assembled (see (*i*) above).

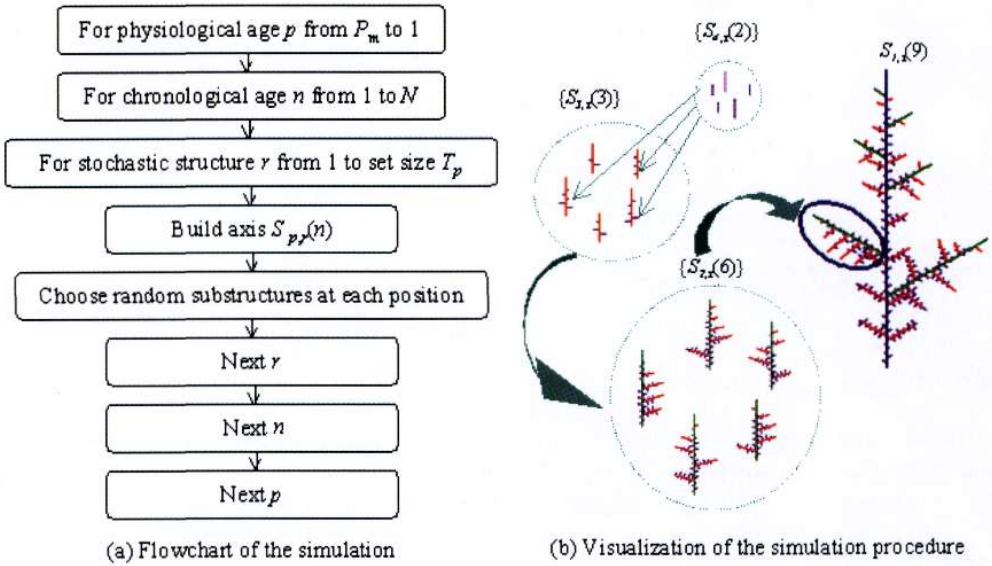
As illustrated in Fig.3a, the simulation thus consists of three nested loops. If the growth is supposed to be fully deterministic, the number of different substructures for a given physiological age is equal to 1 ($T_p=1$ for any p) and the algorithm generates only one instance for a given chronological age and physiological age. On the contrary, when growth is stochastic, there are potentially many different substructures for a given chronological age and physiological age, so that the instancing approach can efficiently be applied: (i) a subset of the physiologically oldest axis is generated according to probabilities p_b, p_c and p_u; this subset is stored into the library of instances; (ii) physiologically younger substructures (e.g. branches) are recursively simulated by assembling physiologically older substructures drawn from the library, according to probabilities p_a and to other automaton parameters; these new substructures are also stored into the library of instances; (iii) the process goes on until the stem of the tree is built (Fig.3b).

3 Results

3.1 Topological production

In fact, plant growth is a compound stochastic process. An interesting result is that the production of organs (e.g. internodes of various physiological ages) can be described by its mean and variance, which can be exactly computed according to the design and parameters of the stochastic automaton. The long computation that gives theoretical results is not to be explained here, but these statistics provide a basis for assessing the performance of our simulation, especially for analyzing the influence of the library size on the predicted outputs.

To ensure that stochastic substructures simulation yields accurate results, we compared the simulated outputs with the theoretical mean and variance of that compound distribution. We choose the same example as in Fig.2 and Fig.3 but set plant age N=24.



(a) Flowchart of the simulation (b) Visualization of the simulation procedure

Fig.3 Description of the simulation algorithm with stochastic substructures. Fig.3(b) illustrates some stages in the simulation procedure based on stochastic substructures. The size of the library of substructure instances is the same for all types of substructures (T_p=T=5), and the maximum physiological P_m is 4. The simulation begins at physiological age 4, with small peripheral unbranched substructures. $S_{p,r}(n)$ denotes the rth instance of substructures of physiological age p when their chronological age equals n. For example, the construction of the subset of substructures $\{S_{3,r}(3)\}$ is obtained by assembling instances of physiologically older substructures that are drawn from the subsets of already simulated substructures (e.g., the bottom growth unit of $S_{3,r}(3)$ bears a branch that is an instance of substructure $S_{4,r}(2)$: this instance is randomly sampled from the 5 instances belonging to subset $\{S_{4,r}(2)\}$

The probabilities are: p_b=0.8, p_c=0.9, p_u=0.8 and p_a=0.5, uniform for each physiological age. If the library of instances is large (i.e., T_p=T=500), the fitting between theory and simulation is good and there is little bias. Practically speaking, even a small library is sufficient to get a good quantitative approximation. For example, we simulated a sample of 100 trees with a small library size: T_p= 5 for physiological ages greater than 1 (Table 1). Despite the small size of the library of instances, the bias remains fairly small, which indicates that it might even be possible to apply instancing more intensively (i.e., to further reduce T_p). Of course, the larger the substructure instance library, the closer between the simulated and real plant productions. On the other hand, we show below that a large library is not necessary if the sole aim is to generate plants that are visually realistic.

It should be noticed that, although the average total production (e.g. measured as the number of metamers) is equal to the sum of each average production for the various substructures, this decomposition does not necessarily hold for the variance, because the covariance between the instances are not null (i.e., as a result of the recurrent construction of

the instances).

Table 1 Simulation vs. theoretical calculation

Age_Ph	*M_S*	*V_S*	*M_Th*	*V_Th*
1	33.08	527.50	33.77	528.22
2	49.32	1167.68	47.17	1051.83
3	71.30	2082.3	67.41	1861.61
4	88.03	4073.52	82.27	2577.81
Total	241.73	23313.08	230.62	20108.26

Age_Ph is the physiological age of organs in tree, *M_S* and *V_S* are the mean and variance of simulated number of organs (i.e. metamers), while *M_Th* and *V_Th* are the corresponding theoretical value.

3.2 Simulation performance

Here we begin to use an example that has more organ production at each age. Fig.4a shows the topological parameters of the stochastic automaton. The following results are based on this automaton with plant age changing from 1 to 30. Our software runs under Matlab, on a PC with an Intel Pentium 4 processor at 1.7GHz, and with a system memory of 256MB.

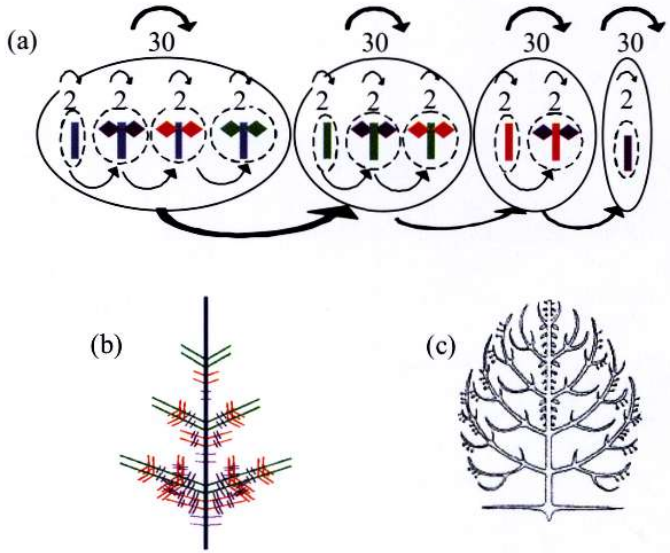


Fig.4 Topological parameters and plant architecture. In Fig.4a, each cylinder is an internode that can bear lateral buds. The numbers indicate the number of repetitions of each state. The arrows indicate the transition probabilities. Fig.4b shows the corresponding structure at chronological age 4 if all the probabilities are set to 1. Fig.4c is a general Rauh model as defined and drawn by botanists: this model can be simulated with such automaton as in Fig.4a

3.2.1 Performance of the simulation for five instances per level

In order to make sure that each bud will be visited, we set all the probabilities to 1. The growth is thus deterministic: this simple case is helpful for checking the performance of the algorithm. The size of the library for each substructure type is 5. Since the growth is deterministic, the instances are all identical for a given physiological and chronological age.

Fig.5 shows the CPU time spent for simulating five trees using the recursive algorithm based on substructures. It took less than 7 seconds at age 30. The total production of organs (i.e., metamers) is listed in Table 2.

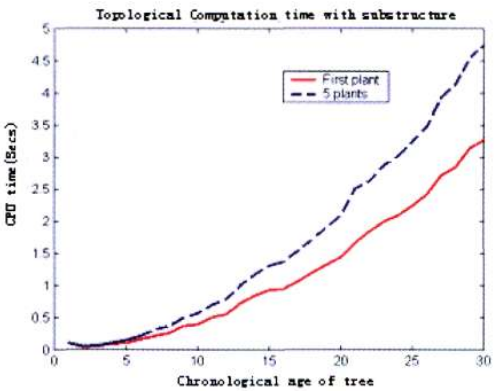


Fig.5 Time spent for topological computation. — time for the computation of the first tree; - - - accumulated time for the computation of the 5 trees

The dashed curve shows the total calculation time for the five trees, and the other shows the time spent for the first tree. It is interesting to see that the simulation required much more time for the first tree, than for subsequent trees: this is natural, because the simulation of first tree includes the construction of the library of instances. For the subsequent trees, the time is spent only for selecting the substructures from the library, and then assembling them at tree level: the process is therefore much faster. It is also worth noting that the gain in simulation time, between the first tree and the subsequent trees, increases when tree age increases, because there are more branches when the trees get older (Fig.5).

3.2.2 Comparison with prefixed order simulation

In this section we compare the performance of the algorithm based on substructure instancing and of a classical "bud-by-bud growth algorithm" that uses the prefixed order simulation. The library size is still 5 for all substructure types, and we simulate 100 trees.

Table 2 lists the results for the first simulated tree and the others: obviously, the prefixed order simulation takes much longer time than the substructure simulation. With prefixed order, trees are simulated independently one by one so that each tree requires the same amount of time. As anticipated, substructure simulation requires much less time than prefixed order simulation, and those for the sebsequent tree is less than those for the first tree. We see that the performance ratio of second tree simulated with substructure to prefixed order simulation is more than 400 at age 30. For 100 trees at age 30, the CPU time spent by the substructure algorithm is about one minute, while the CPU time needed by the prefixed order algorithm is 27, 747 seconds (nearly 8 hours), which would be unbearable for most operational applications.

Fig.6 presents the relationship between simulation time and the total number of tree organs. Fig.6 shows how simulation time increases when the number of simulated plant organs increases: it is worth observing that, with substructure algorithm, this relationship is neither exponential nor linear. We have not yet analyzed this relationship theoretically, but it is sure that it is related with the complexity (i.e., the number of different types of substructures) and total size of the tree. With prefixed order algorithm, the relationship between the number of organs and the computation time is linear: this is normal because each internode is simulated. These results also indicate that the relative efficiency of the substructure algorithm increases when the number of tree organs also increases (see the changes in ratio in Table 2 from age 5 to 30). The dotted line is Fig.6 shows the simulation time for a subsequent tree making use of existing instance library.

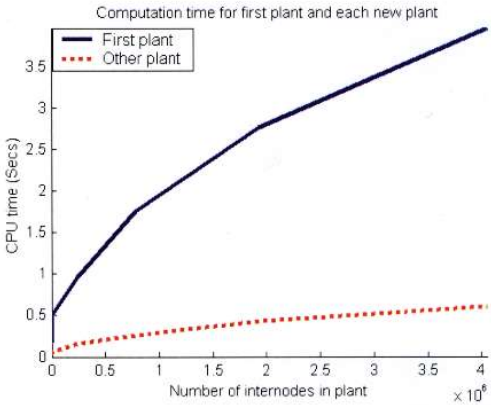


Fig.6 CPU time spent for the computation of 100 trees: the abcissa is the number of organs in the plant. — time for the computation of the first tree; --- computation time for any additional tree

Table 2 CPU time for the computation: comparison of substructure algorithm and prefixed order simulation

N	*Nb_O*	*Prefix*	Sub_1	$Ratio_1$	Sub_2	$Ratio_2$
5	2,440	0.20	0.156	1.28	0.014	14.28
10	4,480	3.14	0.500	6.28	0.056	56.06
15	238,120	17.69	0.953	18.56	0.152	116.38
20	775,360	58.37	1.750	33.35	0.261	223.64
25	1,928,200	132.14	2.766	47.77	0.426	310.18
30	4,048,640	277.47	3.969	69.91	0.603	460.15

N is the number of growth cycles. *Nb_O* is the number of organs (i.e. plant production), which is identical for both simulations. *Prefix* is the CPU time (in second) for prefixed order simulation; Sub_1 and Sub_2 refer respectively to simulation time for first and subsequent trees with substructure method. The ratio $Pre{:}Sub_x$ is noted $Ratio_x$ and measures the relative performance of the substructure algorithm.

3.2.3 Influence of the size of the instance library

It is also interesting to analyze how the size of the library of substructure instances influences the simulation time. Here, we simulate 100 trees which have a chronological age of 30, using a library size that increases from T_p=1 to T_p=200. The solid line in Fig.7 shows the cumulated simulation time for the set of 100 trees, while the dotted one only concerns the first tree: it appears that CPU time increases linearly with set size. The two curves are approximately parallel, the difference being due to the computation time for the subsequent 99 trees: each of which required about 0.6 second (see Table 2), which is the time for sampling the substructures from the library and assembling them; the total difference is thus around 60 seconds.

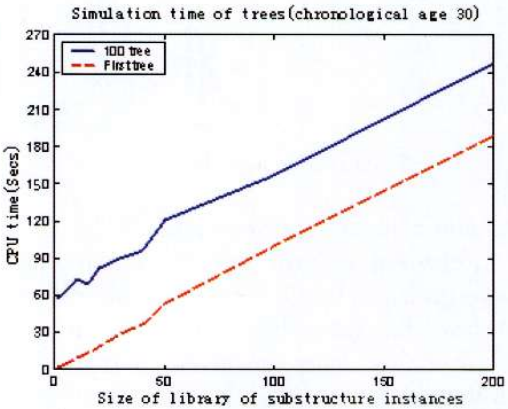


Fig.7 CPU time as a function of the size of the library of substructure instances. --- time for the computation of the first tree; — accumulated computation time for 100 trees

3.3 Graphical results on stochastic substructures

As mentioned in [12], the substructure algorithm can be used not only for the topological computation of the tree, but also for visualizing the 3D structure of plants. For each combination of physiological and chronological age, the 3D substructure instances are stored in the library and can be used by graphical transformation operations (translation and rotations).

As an illustration, we set the library size to 3,4,5 for physiological age 2,3 and 4 respectively, and the probabilities to p_a=[1 0.7 0.7 0.7], p_b=[0.9 0.85 0.8 0.75], p_u=[0.8 0.8 0.8 0.8] and p_c=[0.95 0.93 0.91 0.9] respectively.

Fig.8 (a, b and c) shows 2 random substructures for p=4, 3 and 2 (physiological age), and n=1, 2 and 3 (chronological age). Since the growth of substructure is stochastic, it may happen that some instances are empty (e.g. see the second random substructure instance with physiological age 4 at chronological age 2). Fig.8d shows a simulated tree from age 1 to 4.

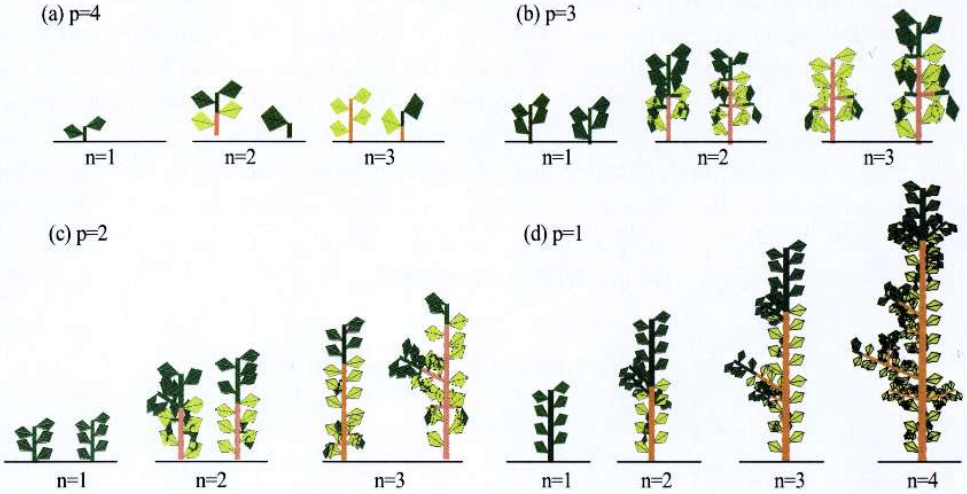


Fig.8 Stochastic plant 3D simulation

3.4 Graphical comparison of stochastic trees and stochastic substructures

It is interesting to compare classical stochastic trees, built using a pure Monte-Carlo method with cruder stochastic trees generated by a limited set of substructure instances. The former are randomly sampled from the theoretical statistical distribution derived from bud functioning probabilities, while the latter are in fact drawn from a sub-sample of this statistical distribution. Considering tree visual display, a small library size can bring nice visual effect: it is even difficult to distinguish the results obtained with different library size, or from the output of a prefixed order simulation.

The minimum library size to get a random tree is obtained with T_p=2 (i.e., 2 instances for each type of substructure). AMAPsim software (written in C language) was adapted to simulate such stochastic trees (i.e., the substructure simulation algorithm was included into AMAPsim). Fig.9 shows two young Plumeria trees simulated with this version of AMAPsim: Fig.9a is the output of the classical bud-by-bud simulation, while Fig.9b represents a tree generated with the substructure technique; the 2 trees look as if they were extracted from the same distribution; no visual impact of the impoverishment is visible in the 3D plant architecture.

Moreover we can compute different trees from different random seed numbers. Using AMAPsim software, the classical Monte-Carlo method generates a single 20-year old tree in 2 minutes, while the substructure method needs only less than one second. So the time gain for this tree is about 200. As a consequence, 600 trees can be produced in 5 minutes when using substructures, while a pure random Monte-Carlo simulation would need more than 8 hours for the same result. This indicates a great advantage of substructures over the classical way. This allows the simulation of planted forest stands in a reasonable amount of time. Fig.10 shows a set of wild cherry trees simulated with Monte-Carlo method, while Fig.11 shows similar results but simulated with substructures.

4 Discussion

Since 1988, we know how to model and simulate stochastic trees, and more generally any type of plants, according to the botanical knowledge of their structure and development[6]. The stochastic functioning of the buds can be analyzed using various methods, and the parameters that control these processes can be estimated on real plants from data measurements[14,15,16]. Secondly, plant architecture and development can be simulated with Monte-Carlo methods[8, 7]: the predicted 3D output of these simulations is visually faithful to real plants and it is a possible to quantitatively check that the predicted tree architecture is similar to the observed one. This model was applied for computer graphics issues, but also in other fields such as agronomy, ecology and forestry[17]: for example, it was used in agronomy for analyzing and modeling light interception by a canopy[18].

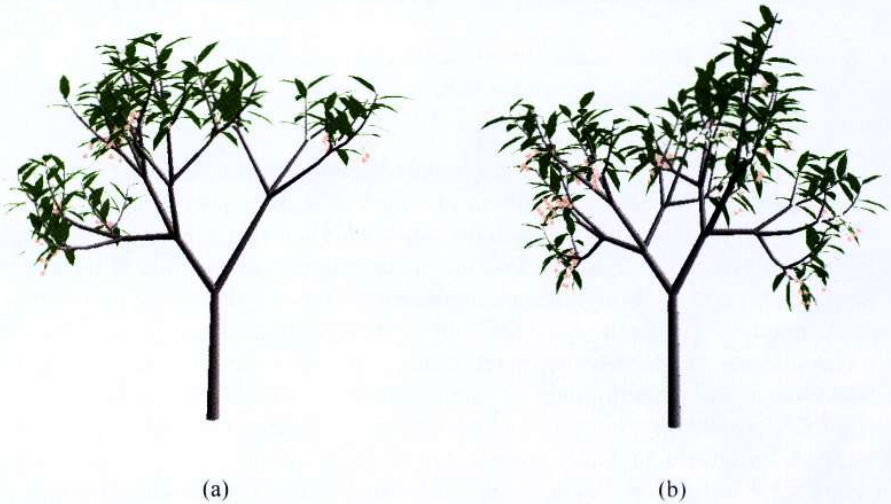


Fig.9 Visual comparison of a Monte-Carlo simulation and of a substructure simulation: simulation of a young Plumeria. (a) Bud-by-bud simulation; (b) substructure simulation with 2 instances

Fig.10 Visual output of Monte-Carlo "bud-by-bud" simulation. Three random 20-year-old wild cherry trees were simulated with the original version of AMAPsim software

Fig.11 Visual output of simulations based on stochastic substructures and instancing. Three 20-year-old wild cherry trees were simulated with a new version of AMAPsim software that uses substructures and instancing techniques

However, a bottleneck remained because of the high number of botanical items in a single plant, which can easily exceed one million of items: as a consequence, the computational time could be as large as several hours for big trees[8] or dense root systems[19]. The breakthrough by Yan et al.[12] is based on the mathematical formalization of the concept of plant substructures and on its algorithmic implementation using instancing techniques. This concept is directly linked to the botanical notion of physiological age, which provides the basis for identifying the parts of the plants that exhibit a similar pattern. To some extent, substructure-based tree growth models exhibit similarities with fractal models[20] that are built with I.F.S. algorithms. However the main difference between the two approaches must be underlined: fractals do not really grow and their self-similarity lays on static geometrical patterns and scale factors; on the other hand, the substructure-based model presented in this paper is based on an analysis of plant growth. As a consequence, the nature and size of the substructures is not only a matter of scaling geometrical parameters, it is mainly the output of both their topological position in tree structure, which is defined by their physiological age, and of the underlying morphogenetic processes, which control their stage of development. A major advantage of this new method is thus that the simulation algorithm lays on a mathematical model that can be directly interpreted by botanists.

In this paper, we show that method of substructure can be extended to stochastic trees, which paves the way for enhancing the simulation of large landscapes which contain a high number of trees. As expected, the simulation time is proportional to the size of the library of instances. It also depends on the size of the plant as measured by its total number of organs, but this relationship is not linear (Fig.6), while it is linear for classical Monte-Carlo methods. Another interesting feature of the substructure-based algorithm is that its relative efficiency increases when the size of the simulated forest (i.e., the number of simulated trees) also increases: the first tree is computed much faster than with a classical Monte-Carlo methods; for subsequent trees, the substructure-based algorithm is even much faster than for the first tree, because the computation of the latter also includes the construction of most, if not all, the library of substructure instances. For plants which have a simple architecture, such as

maize or sunflower, substructure algorithm has no advantage for two reasons: (i) these plants are simple enough to be efficiently simulated with classical algorithms; (ii) the decomposition of these plants into substructures yields only a few, if not only 1, type of substructure.

Thanks to its underlying mathematical formulation, this new model can also be used to derive the theoretical mean and variance of the tree production (de Reffye, personal communication). The comparison of simulated and theoretical plant production indicates that a library size of 5 is sufficient for agronomical or forest purpose (Table 1), whereas a library size of 2 seems to be sufficient to obtain a good 3D realism (Fig.9). There are other applications of the decomposition of a plant into substructures and of their instancing. These techniques can provide a rational method to build simplified clusters for radiosity[21], sets of branches for tree biomechanics[22], or texels and billboards for computer graphics images[23]. According to the type of application, especially in the field of ecology, it would be necessary to adapt the algorithm that is presented here.

Acknowledgements

This research was supported by LIAMA (Sino-French Laboratory for Computer Sciences, Automation and Applied Mathematics), and Natural Science Foundation of China (60073007).

References

1 P. Prusinkiewicz and A. Lindenmayer. The algorithmic beauty of plants. Springer Verlag. 1990.

2 P. Prusinkiewicz. "A look at the visual modelling of plants using L-systems," *Agronomie*. vol. 19, pp. 211-224, 1999.

3 W.T. Reeves and R. Blau. "Approximate and probabilistic algorithms for shading and rendering structured particle systems," *Computer Graphics*. vol. 19, no. 3, pp. 313-322, 1985.

4 P.E. Oppenheimer. "Real time design and animation of fractal plants and trees. *Computer Graphics*, vol. 20(4), pp. 55-64, 1986.

5 X. G. Viennot, G. Eyrolles, N. Janey and D. Arquès. "Combinatorial analysis of ramified patterns and computer imagery of trees," *Computer Graphics*, vol. 23, no. 3, pp. 31-39, 1989.

6 Ph. de Reffye, C. Edelin, J. Françon, M. Jaeger and Puech C. "Plant models faithful to botanical structure and development," *Computer Graphics*, vol. 22, no. 4, pp. 151-158, 1988.

7 Ph. de Reffye, E. Elguéro and E. Costes. "Growth units construction in trees: a stochastic approach," *Acta Biotheoretica*, vol. 39, pp. 325-342, 1991.

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

9 J. F. Barczi, Ph. de Reffye and Y. Caraglio. "Essai sur l'identification et la mise en oeuvre des paramètres nécessaires à la simulation d'une architecture végétale : le logiciel AMAPsim," In: Bouchon J., de Reffye Ph., Barthélémy D. (Eds.), *Modélisation et Simulation de l'Architecture des Végétaux*, Science Update, INRA Editions(FRA), Paris, 1997, pp. 205-254.

10 O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr and P. Prusinkiewicz. "Realistic modeling and rendering of plant ecosystems," *Computer Graphics*, vol. 32, no. 3, pp. 275-286, 1998.

11 J.C. Hart. "The Object Instancing Paradigm for Linear Fractal Modeling," In: *Proc. of Graphics Interface 92*, 1992, pp. 224-231.

12 H.P. Yan, J.F. Barczi, Ph. de Reffye and B.G. Hu. "Fast Algorithms of Plant Computation Based

on Substructure Instances," In: *WSCG' 03 Conf. Proc*, 2002, vol. 3, no. 10, pp. 145-153.

13 X. Zhao, Ph. de Reffye, F.L. Xiong, B.G. Hu and Z.G Zhan. "Dual-scale automaton model for virtual plant development," *The Chinese Journal of Computers*, vol. 24, no. 60, pp. 608-617 (Chinese, with an English abstract). 2001.

14 C. Godin and Y. Caraglio "A multiscale model of plant topological structures," *Journal of Theoretical Biology,* vol. 191, pp. 1-46, 1998.

15 Y. Guédon, D. Barthélémy, Y. Caraglio and E. Costes. "Pattern analysis in branching and axillary flowering sequences," *Journal of Theoretical Biology*, vol. 212, pp. 481-520, 2001.

16 P.Heuret, D. Barthélémy,Y. Guédon, X. Coulmier and J. Tancre."Synchronisation of growth, branching and flowering processes in the South America tropical tree *Cecropia obtuse* (Cecropiaceae) ," *American Journal of Botany*, vol. 89, no. 7, pp. 1180-1187, 2002.

17 Ph.de Reffye and F. Houllier. "Modelling plant growth and architecture: some recent advances and applications to agronomy and forestry," *Current Science*, vol. 73, no. 11, pp. 984-912, 1997.

18 J. Dauzat, and M.N. Eroy. "Simulating light regime and intercrop yields in coconut based farming systems," *European Journal of Agronomy*, vol. 7, pp. 63-74, 1997.

19 C. Jourdan and H. Rey. "Modelling and simulation of the architecture and development of the oil-palm (Elaeis guineensis Jacq.) root system," *Plant and Soil*, vol. 190, pp. 217-233, 1997.

20 A.R. Smith. "Plants, fractals and formal languages," *Computer Graphics*, vol. 18, no. 3, pp. 1-10, 1984.

21 C. Soler, F. Sillion, F. Blaise and Ph. de Reffye. "An efficient instanciation algorithm for simulating radiant energy transfer in plant model," *ACM Transaction on Graphics*, vol.22, no.2, pp.204-233

22 T. Fourcaud, F. Blaise, P. Lac, P. Castera and Ph. de Reffye. "Numerical modelling of shape regulation and growth stresses in trees PART II: implementation in the AMAPpara software and simulation of tree growth," *Trees, Structure and Function*. 2002.

23 A. Meyer and F. Neyret. "Multiscale Shaders for the Efficient Realistic Rendering of Pine-Trees," *Graphics Interface*, pp. 137-144. 2000.