INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A physiological plant growth simulation engine based on accurate radiant energy transfer

Cyril Soler — François Sillion — Frédéric Blaise — Philippe de Reffye

N° 4116

Fevrier 2001

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4116--FR+ENG

A physiological plant growth simulation engine based on accurate radiant energy transfer

Cyril Soler , François Sillion , Frédéric Blaise , Philippe de Reffye

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projets iMAGIS* et LIAMA†

Rapport de recherche n° 4116 — Fevrier 2001 — 31 pages

Abstract: We present a new model for plant growth simulation, taking into account the eco-physiological processes driving plant development with unprecedented fidelity. The growth model, based on a physiological analysis, essentially simulates the internal function of the plant, and has been validated against measured biological data with excellent results. We show how to account for the influence of light through photosynthesis, and thereby incorporate the effects of a given plant's immediate environment on its architecture, shape and size. Since biological matter is controlled by water transpiration and received radiant enery, the model requires efficient and accurate simulation of radiant energy exchanges. We describe a complete lighting simulation system tailored for the difficult case of plants, by adapting state-of-the-art techniques such as hierarchical instanciation for radiosity and general BRDF modeling. Our results show that (a) our lighting simulation system efficiently provides the required information at the desired level of accuracy, and (b) the plant growth model is extremely well calibrated against real plants and (c) the combined system can simulate many interesting growth situations with direct feedback from the environment on the plant's characteristics. Applications range from landscape simulation to agronomical and agricultural studies, and to the design of virtual plants responding to their environment.

Key-words: Plant growth simulation, physiological simulation,
lighting simulation, radiosity, instanciation,
landscape simulation, calibrated physiological simulation

* iMAGIS est un projet commun entre CNRS, INRIA, INPG et UJF
† LIAMA: Laboratoire Franco-Chinois d'Informatique, d'Automatique et de Mathématiques Appliquées, Beijing, Chine.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Un moteur de simulation physiologique de croissance des végétaux basé sur le calcul des transfers radiatifs

Résumé : Ce rapport présente un nouveau modèle de simulation de la croissance des végétaux prenant en compte les processus physiologiques responsables du developpement des plantes, avec une précision non encore égalée. Ce modèle de croissance est basé sur la collaboration entre un moteur simulant le fonctionnement interne de la plante, ayant été validé expérimentalement avec succès, et un algorithme de simulation de l'éclairage. La production de matière végétale au niveau des feuilles dépend en effet, par l'intermédiaire de la quantité d'eau transpirée, de la quantité de lumière qu'elles reçoivent. Son calcul demande donc une estimation précise de l'energie lumineuse reçue par les feuilles. Pour l'obtenir, nous proposons un algorithme d'éclairage global adapté au cas difficile des scènes végétales, faisant intervenir des techniques de radiosité hiérarchique avec *clustering*, d'instanciation, et d'analyse statistique de fonction de réflectance bidirectionnelles. Nos résultats montrent que (a) notre système effectue ce calcul de manière efficace au niveau hiérarchique requis, (b) le modèle de croissance des végétaux est particulièrement bien callibré comparativement à des scènes réelles, et (c) la combinaison du calcul de l'éclairage et du moteur de croissance permet de simuler plusieurs phénomènes où l'environnement d'une plante agit, par l'intermédiaire de la lumière, sur sa géométrie. Les applications possibles comprennent la simulation réaliste de paysages, l'optimisation de paramètres de culture en agronomie et la conception de modèles de végétaux géometriquement adaptés à leur environnement.

Mots-clés : Simulation de la croissance des plantes, simulation physiologique, simulation de l'éclairage, radiosité, instanciation, paysagisme

1 Introduction

The creation of realistic plant models is required by many applications in very diverse fields. In computer graphics, we have seen many examples of beautiful renderings of trees, flowers and other plants, which tremendously add the the realism of virtual scenes. Plant models however are very complex, making it tedious to create them by hand. On the other hand, *Simulating* plant growth offers the fascinating perspective of being able to build, study and render virtual plants in their specific environment, as well as following and controlling all stages of their development.

Possible applications of plant growth simulation include architectural planning, landscape and urban design (to simulate the evolution over time), virtual world design (to obtain realistic plants in their environment), and agronomical science and forestry (to monitor, control and optimize crops and plantations). All these applications share to some degree a requirement for realism that goes beyond appearance and can include proper modeling of competition for space, orientation towards light, reaction to the radiative environment, calibrated reaction to available water, temperature and light and prediction of quantitative data (weight, size of the plant).

Such precise simulation is made particularly difficult by the complexity of underlying physiological processes operating at very different time scales at the biologicallevel. However a key observation is that light is the most important factor to account for in the growth of plants [MP96]. The distribution of radiant energy influences the growth of a plant in at least three ways: (1) since water transpiration is directly linked to the received energy, the production of vegetal matter varies throughout the plant as well as the stimulation for the growth of each organ. This explains many architectural and shape variations in plants, such as those shown in Figure 1.a. (2) during their growth the plant axes tend to aim towards the light sources to maximize the amount of light energy received by their leaves. This phenomenon is illustrated in Figure 1.b. (3) Infrared light has a direct impact on temperature which, in turn, influences the speed at which the plant grows.

In this paper, we present a novel growth simulation engine based on the internal functions of the plant, and show how computer graphics techniques can be adapted to drive the simulation and provide all the required information about light distribution. The combination of eco-physiological laws and proper light simulation allows precise modeling of the modification of the plant's architecture, shape and size throughout the simulation, and in response to external factors.

Clearly a key ingredient of the system is the lighting simulator. Unfortunately the distribution of light in a plant is highly non-uniform due to diffusion, self-shadowing, and indirect lighting. These phenomena must all be simulated to obtain meaningful intensity and lighting direction information for each growing structure of the plant. Thus we have two conflicting goals. We need a method that works at many scene scales: a single plant as well as a large number of trees; and we want accurate predictions at the finest scale, i.e. for each individual leaf. As we shall see none of the existing lighting simulation techniques allows a proper computation in plants, accurate enough to permit the use of a physiological simulation (either for scale limitations, strong approximations or lack of indirect illumination). We present an integrated lighting simulation algorithm tailored for plants, and carefully discuss the design choices pertaining to the degree of accuracy needed and the method used for simulating light propagation in vegetation.

Figure 1: Examples of plant deformation due to light. The right-hand plant has been rotated at regular intervals during growth.

Our plant growth simulator models and predicts internal parameters of the plant such as vegetal matter production and internal wood constraints and is, to our knowledge, the first to offer a calibrated physiological response to environment conditions. Once coupled with a lighting simulator, it automatically models the influence of external constraints (changes in lighting conditions, competition with other plants. . .).

The simulation engine is not geared towards the production of very fancy geometric models for producting impressive images, but this can naturally be done at a later stage by providing highly detailed models of plant organs, using one of the many excellent systems available. Instead our engine provides a well-grounded model of the plants architecture and shape, useful in a wide range of applications: in Computer Graphics, to create models of individual plants responding to their location and environment; in landscape simulation or forestry planning, to adapt plants to their environment and study long term evolution of trees; for agronomy, to optimize plant growth, crop yield, and as a scientific investigation tool that can serve to study other biological models.

Figure 2 summarizes the architecture of the system. The plant growth simulator kernel on the left-hand side is in charge of three tasks: computing the plant topology [dREF$^+$88], computing the volume of organs, and computing the geometry of the plant. In addition to the lighting simulation module, other modules exist but are not discussed here, such as a mechanical simulator to deal with internal mechanical constraints when inferring the geometry of branches.

After reviewing relevant work in the next section, we devote Section 3 to the explanation of the physiological plant growth simulator and present its calibration. Section 4 then presents the lighting simulation algorithms, emphasizing and justifying the important design choices made for the specific case of plants.

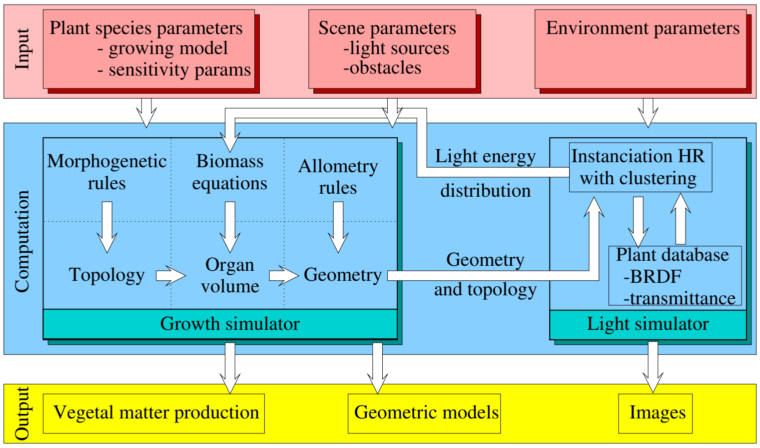


Figure 2: Architecture overview of our plant growth simulation system. See text for details.

2 Previous work

Plant growth simulation

The 3D simulation of plant growth and plant architecture to date has been mainly morphogenetic and uses algorithms such as computer grammars [PLH88, PHM93] or stochastic processes [dREF^{+}88], that simulate the genetic program monitoring the plant development. The “architectural models" of trees [HOT78] are good examples. In such a case the morphogenesis is only concerned with geometric plant description, i.e. the relationship between the botanical items stacked in a topological structure.

Consequently the sizes of botanical entities in the 3D plant model (e.g. leaves, or branch elements called inter-nodes) have to be adjusted from direct observations (this is especially the case for the nice looking plants of [WP95]) and not from a functional computation of the photosynthesis. In essence the plant is dressed after the topological structure is computed.

New models have recently been introduced to simulate the interaction of plants with their environment [MP96], including other neighbouring plants. This is achieved by using hand-made rules in order to visually reproduce the reaction of plants to factors such as light intensity or competition for space.

Therefore in this model the organs do not play a functional role and are only set automatically in place during the growth. There is no reliable information about the biomass produced nor about the speed of growth. In other words, the plant is not treated as a living entity but rather a geometric object whose geometry is adapted to its environement. Similarly, the impact of external conditions such as lighting conditions can be estimated by perturbing the geometry [Han95].

A new trend is based on the recognition that plant structure is the joint output of the physiological processes (water and carbon balance, etc.) and the morphogenetic program of the plant. Such issues, involving vegetal matter fabrication by estimating photosynthesis and matter transfer inside the plant architecture has been tackled recently [Jal98] with software packages such as Lignum [PSN+96], Grogra [Kur94], or AMAPhydro [dRFB+96, BBJ+98]. In these systems the botanical organs begin to play their actual functional role inside the plant structure.

Clearly there are several types of applications for virtual plants, with different requirements on the algorithms employed. In the context of computer graphics a fairly arbitrary set of rules can be used to govern growth and the interaction of the plant with its neighbors, as long as the results are visually satisfactory to the user. The best system to date based on user-defined rules is presented in [MP96]. Despite their very attractive appearance (undoubtfully satisfying for visual computer graphic applications) the results cannot be validated against reality because of the independent, uncalibrated behavioral models employed.

On the other hand, the challenge of growing virtual plants faithful to agronomical laws, in a manner that can be validated by comparing a simulation to measured data, is fascinating because of its many useful applications. An example would be yield optimization in various configurations of plantations, taking into account plant growth and architecture. Virtual plants, once successfully validated by a sufficient agreement between simulation and measurements, can be used for virtual simulation of plantations, urban simulation, resources optimisation, etc. Such work has been carried out recently, making it clear that the way of measuring the real plant, and of computing the hidden parameters of the model can be a mathematical issue at least as difficult as the simulating process itself [dREC91, dRH97, GGCC97, dRBC+99].

The simulator described in this paper extends the resulting models, integrating biological, ecophysiological and physical laws in the development of the plant model and computing their combined results.

Simulating light energy in plants

Concerning plant growth simulation, we are interested in the class of methods that compute the amount of light received by leaves of the plants.

A number of methods estimate direct illumination in the plant possibly using attenuation factors, but without accounting for light scattering inside the vegetation. A straightforward approach is to perform ray casting toward the sky and through the model, like Perttunen *et. al* [PSN+96] or through a voxel representation of it as in Greene [Gre89]. Měch *et. al* extended this technique by accumulating the opacity of voxels successively encountered by a ray to account in a way for translucency of the foliage. These first three approaches do not account for light scattering inside the model, due to the essentially diffuse transparency of the leaves [Gov95]. with a definite impact on the illumination, and therefore on growth and architecture (see Section 5, Moreover, in the last model, voxel opacity is not computed from its real content but simply fixed to an arbitrary value. However, these approaches are sufficient for determining coherent values and directions of illumination for non physiologicaly-based plant growth.

Global illumination hase also been used to compute light energy distribution in plants. Max [MMKW97] proposes a simplification of the radiant transfer equations in order to compute a density of light for each altitude in the canopy. This economical but realistic approach is only suitable for large scale plant models including some repetition, because of the isotropy assumptions made on the model. It is therefore only applicable to large plantations and not to small-scale scenes.

Radiosity techniques are usually quite costly although they produce faithful results. Many approximations of the radiosity method have been proposed up to now: Goel *et. al* [GRT91] obtain a radiosity solution in a corn field using periodicity assumptions, which reduces the number of form factors to compute to that with the neighboring polygons of each plant. The use of standard radiosity on a pure geometric model limits this approach to scenes with a small number of polygons. Borel *et. al* [BGP91] propose to set form factors to distant objects to 0. This introduces a bias in the solution but enables to handle large scenes. Chelle *et. al*'s *nested radiosity* algorithm uses a geometric model for the local neighborhood of a polygon and a volumetric model using scattering equations for distant geometry [CAB98]. This requires isotropy assumptions on distant parts of the canopy and periodicity of the model. This work is also only applicable to large-scale scenes like plant canopies.

None of these radiosity simplifications can adapt to a wide range of scene scales because they do not use hierarchical approaches and partly because they need isotropy assumptions in the model. This may explain why none of them was ever applied to plant growth simulation.

Monte Carlo methods have been used by Ross *et. al* [JA88] and Govaerts [Gov95] to estimate the canopy bi-directional reflectance function. These techniques work well for BRDF computation because they do not need to save the distribution of light inside the model. Dauzat *et. al* [DM87] use it to estimate the light received by the leaves of plants taking into account internal light scattering. Like all stochastic methods, Monte Carlo approaches have two drawbacks: they converge very slowly and the accuracy of the result is not easily controlable. Vegetation indeed contains very uncorrelated polygons, and thus induces a large dispersion of rays hence an especially noisy and slow convergence, making them poorly suitable to an interactive work in hand with a plant growth engine.

3 Physiologically based growth model

We introduce a model of plant development based on the simulation of physiological processes taking place inside the plant organs, such as water evaporation (transpiration), synthesis of fresh vegetal matter, and allocation of the fresh matter in the organs during their growth.

The crucial difference between this model and other plant simulation techniques based on automata and rules lies in the precise description of the internal matter allocation system. This mechanism is of course controlled by parameters, representing the strength with which different organs compete for matter allocation: these parameters cannot be directly measured on real plants, but the model can still be calibrated in a global manner using parameter estimation techniques to recover the hidden parameter values. Such calibration is presented later in the paper, and is a major strength of the approach. Since the simulation is based on the internal functioning of the plant, growth and various geometrical shape deformations due to light non-uniformity are automatically obtained as "side effects".

3.1 biological assumptions

The virtual plant is assumed to be made of fresh matter, containing about 80% of H_2O and 15% of carbohydrates (CO_2 + H_2O). Overall water represents 95% of the plant. It is safely assumed that water controls all matter allocation in the architecture. The simulator only operates on the aerial part of the plant, whose weight is assumed proportional to that of the root system. Our simulator is based on an architectural description of plants in terms of *organs*, which can be inter-nodes, fruits, leaves etc. The architecture of a given species is expressed in transformation laws as in [dREF$^+$88]. The originality of the current model lies in the computation of physical parameters of the organs, such as size and weight, according to physiological processes. This move from an abstract representation to a physiologically grounded one is made possible in particular thanks to the following two agronomical laws:

Law of the sum of temperatures Plants grow in cycles (the duration of which ranges from several days to a year). Sequences of growth cycles implicitly form the basis of all existing plant simulators, in the form of operations or transformations in automata and grammars. However when attempting a faithful physiological simulation, the duration of these cycles should be taken into account. For instance a cotton tree has cycles ranging from 3 to 10 days during the summer. Fortunately *the number of cycles undergone by a bud in a given time range is proportional to the sum of the daily average of the temperatures*. Therefore the same plant can be grown at different speeds under different climates. This law determines the number of cycles for a real time period of growth for a given plant, and allows us to continue reasoning on unit cycles, or thermally adjusted cycles. Note that the distinction between thermal time and real time may seem subtle for tempered trees with a yearly growth cycle, but becomes very important for tropical plants, for instance.

Law of the water use efficiency Over a sufficient duration and subject to a given amount of incoming light, *the amount of fresh matter fabricated is proportional to the water evaporation of the plant*. The proportionality factor is called the *water use efficiency*, and varies depending on the biological type of the plant. For instance it is known that about 600 liters of water are needed to produce 10 kg of fresh matter of potatoes. We use this proportionality to derive the amount of matter produced from the computed plant transpiration, and also assume that there is no hydric stress.

These two laws are extremely well verified by experiments on a wide variety of plants, as indicated in Figure 3, and form the physiological basis for basing the simulation on radiant energy computations.

3.2 Overview of the simulation

For the purpose of explanation, each growth cycle is decomposed into three functional stages, which actually correspond to different periods in the cycle for most plants:

First, the available reserve of fresh matter is used to create new organs or to carry out the expansion of existing ones. The number of new organs depends on the specific architectural model of the plant at hand, which defines its topological structure [dREF$^+$88]. The geometry of the plant is computed according to intrinsic allometry rules (constraining the shape of the organs), as well as behavior rules such as photo-tropism (the capacity of the plant to align its organs towards the principal direction of illumination).

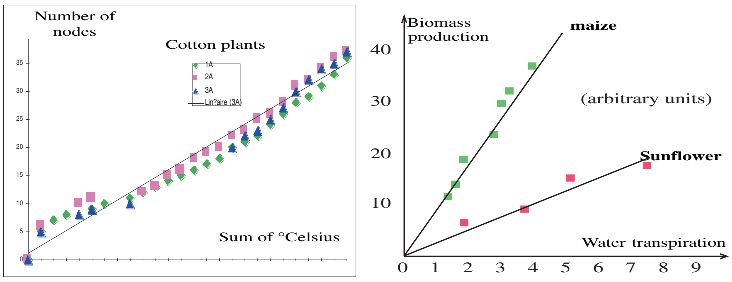


Figure 3: Experimental verification of agronomical laws.

Second, the architecture remains fixed while the effect of photosynthesis is computed, in the form of fresh matter creation. During the cycle, the architecture remains fixed and the leaves perform photosynthesis.

At the end of the cycle, secondary growth is carried out, creating rings inside the trunk and branches, and growing fruits. The remaining fresh matter is put in reserve for the beginning of the next cycle where it will be allocated to the new buds.

3.3 Modeling ecophysiological processes

In order to carry out the simulation of our three stages, mathematical models are needed to express both the creation of fresh matter for a given architecture, and the subsequent allocation of this fresh matter within the plant.

3.3.1 Computation of fresh matter production

Fresh matter is created as a byproduct of sap circulation in the plant. Water is drained from the ground by the roots and flows through the axes into the leaves where it is partially evaporated. The distribution of this flow obeys simple network rules based on the hydric resistivity of all parts of the plant.

During a cycle at time t the total quantity of water evaporated by the plant is given by:

$$Q(t) = \frac{\Delta\Psi(t)}{R(t)}\Delta t$$

where Δt is the duration of the cycle, $\Delta\Psi(t)$ is the difference of hydric potential (partial water pressure) the plant is subjected to, and $R(t)$ is the global hydraulic resistance of the plant, which depends on its architecture and geometry.

Introducing the *water use efficiency* $W_e(t)$, we obtain the weight of biomass fabricated as: $Q_m(t) = W_e(t)Q(t)$. A major recent advance in physiological modeling has consisted in identifying the product $E_m = W_e(t)\Delta\Psi(t)\Delta t$ as the *potential of fresh matter fabrication* (units of weight), and observing *in situ* that it can be considered constant, because the expansion of organs over several growth cycles absorbs its observed random oscillations[dRBC+99]. This assumption is valid for optimal conditions (uniform illumination, no hydric stress) as a first approximation, and links the duration of growths cycles to the external conditions. From that we immediately get

$$Q_m(t) = \frac{E_m(t)}{R(t)}$$

Modeling hydric resistances Typical values for hydraulic resistance (in arbitrary units) would be

$$\begin{cases} \text{ground-root interface} & 5 \\ \text{architecture (axes, leaves)} & 25 \\ \text{leaf-atmosphere interface} & 800 \end{cases}$$

It is therefore valid to assume that the major hydraulic resistance of the plant lies in the leaves, and to model the plant in a first approximation as its set of active leaves operating in parallel. The hydraulic resistance of a single leaf is modeled after Darcy's law by the following formula (serial setup of blade and petiole):

$$R = \frac{\rho_1 e}{S} + \frac{\rho_2 l}{s}$$

where S is the leaf surface area, e the leaf thickness, l and s the length and cross-sectional area of the petiole, ρ_1 the hydraulic resistivity of the blade and ρ_2 the resistivity of the petiole. Allometry rules usually enforce a fixed thickness e for leaves of all sizes, and a fixed l/s ratio for petioles. Therefore the hydraulic resistance of a leaf can be modeled in reduced units (areas in cm^2)as

$$R = \frac{r_1}{S} + r_2$$

where r_1 is the leaf blade resistance per unit area (note the inverse variation of resistance with area), and r_2 is the average resistance of the network constituted by the nerves and petiole of the leaf (typical resistivity values are about $r_1 = 800,000$ and $r_2 = 0.96$).

Amount of fresh matter produced During the cycle t, matter production only depends on the number of leaves $N(t)$ that are available for photosynthesis. This number is a result of the architectural model. If each leaf receives the same amount of energy and undergoes the same climatic effect per unit area on the leaf surface, the biomass production at cycle t is:

$$Q_m(t) = \sum_{x=1}^{N(t)} \frac{E_m}{\frac{r_1}{S(x,t)} + r_2}$$

A more precise expression is obtained in the case where each leaf receives a different amount of energy during the cycle:

$$Q_m(t) = \sum_{x=1}^{N(t)} \frac{E(x,t)}{\frac{r_1}{S(x,t)} + r_2} \tag{1}$$

In the above formula, $E(x,t)$ is the potential of matter production of the x-th leaf at the t-th cycle of growth . It is typically modeled by the formula [dRBC+99]:

$$E(x,t) = \begin{cases} E_m \frac{\Phi(x,t)}{\Phi_m} & \text{if } \Phi(x,t) < \Phi_m \\ E_m & \text{if } \Phi(x,t) \geq \Phi_m \end{cases}$$

In other words, the efficiency of a leaf if proportional to the incoming radiant energy $\Phi(x,t)$, as long as this quantity remains less than a threshold Φ_m ($\approx 1{,}000\, W/m^2$). This is a suitable simplification of the real behavior of leaves [dRBC+99]. These formulae are used to interface the architectural growth model with a model of radiative transfer provided by the radiosity algorithm. The accurate computation of $\Phi(x,t)$ will be treated in section 4.

3.3.2 Allocation of fresh matter in the architecture

Let us now consider how the fresh matter produced can be distributed within the plant. One could naively think that matter is used locally around the organ that has produced it, but in fact water flows very well in the plant, and matter travels everywhere (down to the tip of the roots!). A more flexible and physiologically valid model consists in assigning to each organ a *sink* of a given strength, i.e. a specific demand for matter, that it needs to sustain its growth in optimal conditions.

Our distribution model consists in allocating the globally available matter in proportion of each organ's demand (therefore implicitly allowing free movement of matter in the plant through sap circulation). Note that although two separate steps of matter distribution were discussed above (one at the beginning of the cycle to set up the new architecture, one at the end of the cycle to grow rings and fruits), we can consider a single operation both mathematically and algorithmically. Care must be taken however to organize the competition for matter distribution using some organs from the N^{th} cycle (trunk rings/fruits) and some organs of the $N+1^{th}$ cycle (buds/leaves/inter-nodes).

Note that botanical elements can be added (by the architectural development) or removed (by natural death or pruning) at each cycle, with an automatic balancing of the allocation system. Therefore growth is actively correlated with the architecture and the effects of human intervention. We shall see below an example of this retroaction for pruned cotton trees.

For organs of a given category O, we define their sink strength as P_o and the number of organs in the category at cycle t as $N_o(t)$. The main difficulty in defining allocation rules arises from the fact that organs grow and function over several cycles, with different demand through time. To account for this deferred expansion, we model for each organ a normalized expansion function $F_o(i)$ defining the expansion capacity of an organ of category o with an age of i cycles. We have by definition $\sum_i F_o(i) = 1$. The total demand in matter is given by

$$D(t) = \sum_o P_o \sum_{i=1}^{t} F_o(i) N_o(t-i+1)$$

and the increase in volume[1] for an organ of age i is

$$\Delta q_o(i) = \frac{P_o F_o(i)}{D(t)} Q_m(t)$$

The simple case of immediate expansion, where organs are created directly with their final size, is modeled by setting $F_o(i) = \delta(i-1)$. This case applies well to tempered trees which create their leaves over the course of a few days in the spring.

3.4 Discussion of growth patterns obtained from the model

The application of our physiological simulation model can be expressed mathematically using generalized induction rules. It is interesting to conduct a case study on simple examples, as it provides useful insight into the general growth patterns, extremely well correlated with observations.

Let us first consider a simplistic plant with a single functional leaf on its terminal axis, whose stem has no rings, and fruitless, subjected to uniform lighting conditions. Fresh matter produced at cycle n is given by

$$Q_m^{(n)} = \frac{E_m}{\frac{r_1}{S^{(n)})} + r_2}.$$

The area of the leaf blade is obtained by the allometry constraint (fixed thickness e) from the amount of matter $Q_{ma}^{(n)}$ allocated to the leaf at the beginning of the cycle, as

$$S^{(n)} = \frac{Q_{ma}^{(n)}}{e}$$

We have a single inter-node and a single leaf to create at each cycle of growth, of respective sink strengths P_i and P_l, therefore the biomass allocated to the leaf is linked to the matter produced in the previous cycle by

$$Q_{ma}^{(n)} = \frac{P_l}{P_i + P_l} Q_m^{(n-1)}$$

Putting it all together we obtain the induction rule

$$Q_m^{(n)} = \frac{E\ Q_m^{(n-1)}}{A + B Q_m^{(n-1)}} \tag{2}$$

with [2]

$$A = \frac{r_1(P_i + P_l)e}{P_l}, \quad B = r_2, \quad E = E_m$$

[1] Note that volume and weight are “equivalent" measures here since the plant is made almost entirely of water, with density 1.

[2] to simplify the discussion we assume that all relevant quantities have been reduced to unitless variables using appropriate ratios.

Equation 2 is an homographic sequence, and has the following solution, using $K = \frac{A}{E}$ and $Q_m^L = \frac{E-A}{B}$:

$$Q_m^{(n)} = \frac{Q_m^L}{1 - K^n \frac{Q_m^{(0)} - Q_m^L}{Q_m^{(0)}}} \tag{3}$$

which means that, starting from a seed $Q_m^{(0)}$, the biomass produced per cycle increases if $K < 1$, and reaches the limit Q_m^L, as illustrated in Figure 4. Note that B controls the relative resistance of the petiole compared to the blade, and has a strong impact on the amount of matter produced.

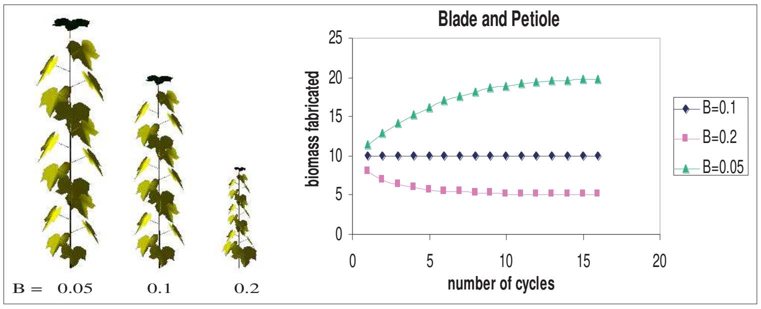


Figure 4: Variation of per-cycle matter production as a function of petiole resistance, for a simplistic plant.

This overly simplified model serves mainly for illustration purposes. However it is easily extended to the case of plants producing $N_a^{(n)}$ identical phytomers at cycle n, providing an induction rule of the form :

$$\frac{Q_m^{(n)}}{N_a^{(n+1)}} = \frac{N_a^{(n)}}{N_a^{(n+1)}} \frac{E \frac{Q_m^{(n-1)}}{N_a^{(n)}}}{A + B \frac{Q_m^{(n-1)}}{N_a^{(n)}}}$$

where the ratio of the numbers of active leaves in successive cycles, $C_n = \frac{N_a^{(n)}}{N_a^{(n+1)}}$, plays a direct role. In general C_n reaches a limit as n increases, typically $C = 0.5$ for a binary tree, or $C = 1$ if a stable canopy forms (e.g. by pruning). Under growth conditions ($\frac{CE}{A} > 1$), important architectural considerations can be deduced: For $B = 0$ (all resistance in the blade area), biomass grows exponentially as

$$Q_m^{(n)} = \left(\frac{E}{A}\right)^n Q_m^{(0)} \quad ,$$

and for $B > 0$, the system is stabilized by the petiole resistance, and the per-leaf biomass production asymptotically tend towards the limit

$$\frac{Q_m^{(n)}}{N_a^{(n+1)}} \rightarrow \frac{CE\ -\ A}{B}$$

Analyzing this formula shows that the more ramified the architecture, the smaller the organs will be, a property verified on real plants.

These induction formulae can be fully generalized for the case of deferred expansion (organs transform fresh matter differently over their life span), to yield a finite difference equation

$$Q_m^{(n)} = F(E_m, N_a^{(n)}, \ldots, A, B, C, \ldots, Q_m^{(n-1)}, \ldots, Q_m^{(k)}, \ldots, Q_m^{(0)}) \tag{4}$$

where $A, B, C\ldots$ are synthetic parameters depending on allometry, sink strengths and organ hydraulic resistances. This equation, describing the interaction between the plant architecture, the fresh matter production and the hydraulic and allometric characteristics of the species, provides a number of general rules for plant growth. These generic predictions, summarized below, match the observed development of almost all plants :

- Growth is exponential during the youth, and the organs increase in size.
- Growth is stabilized after several cycles because the hydraulic constraints are mainly located in the leaves. Organs reach a limit size. The growth in height becomes linear and the growth in weight depends on the architectural model: the more ramified the tree is, the more matter is produced but the smaller the organs.
- Growth eventually slows down if the hydraulic resistance of the axes becomes significant (case of large trees or case of cavitation – when air appears in the branches).

3.5 Validation of the NWFA model

The NWFA[3] model presented here has received very good calibration on cotton and wheat plants.

3.5.1 Qualitative validation: trimmed cotton tree

Figure 5 shows a comparison (performed at our institute) of free-growing and pruned cotton trees. The spectacular effect of pruning is to significantly increase the size of organs, yielding an overall lighter tree. Note that the model reproduces these effects, which can be interpreted by the fact that more fresh matter is available for each of the few remaining organs after pruning.

[3]Name withheld for anonymity

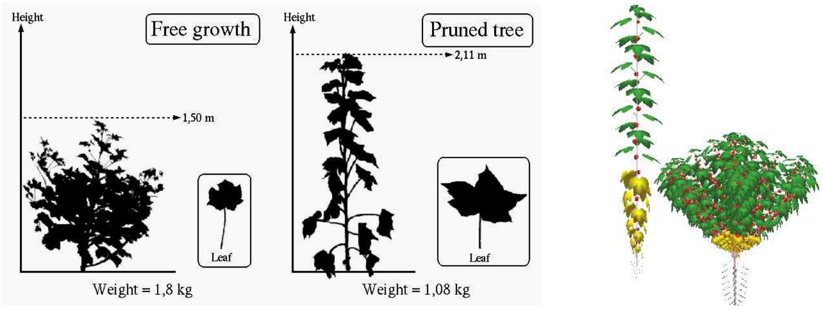


Figure 5: Reaction of a cotton tree to pruning, and result of a simulation (right)

3.5.2 Quantitative validation: identification of hidden parameters

For quantitative validation we need to provide numerical values for all of the model parameters (sink strengths, expansion functions and resistances). These parameters cannot be directly measured on plants, but their effect is nevertheless accessible on transformed variables (weights, areas, lengths, diameters...). These measurable variables are used as targets in a parameter estimation process using any of the well-established techniques (adjoint operators, simulated annealing or least-squares methods). The formulation of our model as a finite difference equation makes generalized least squares estimation very efficient: a dozen of parameters are computed with great accuracy in a few iterations. The advantage of using a least-squares approach lies in the simultaneous production of an error estimate for each parameter.

Cotton tree Figure 6 shows results given by the full model on pruned cotton tree, where leafs expand over about 15 cycles, and inter-nodes over 5 cycles. Note the excellent agreement with measured data We emphasize that the use of a parameter estimation technique to obtain hidden parameters is not sufficient to obtain a good fit, but rather underscores the quality of the model and its ability to predict the correct behavior of all observable variables. All measured plants have been correctly matched by the model, with very little variation in the recovered parameters, which testifies to the robustness of the model.

Wheat We performed similar experiments on wheat, which is quite different in that it has a finite growth, without a steady state or matter production. It also has a fruit. Again (see Figure 7) the match is excellent with almost no variation of the parameters. Furthermore, parameters that can be compared between cotton tree and wheat have similar orders of magnitude.

3.6 Description of a simulation step

We briefly review here the simulation algorithm to provide a global view of the software system. The data structure representing the plant is a hierarchy of *axes*, *growth units*, and *inter-nodes* defining the

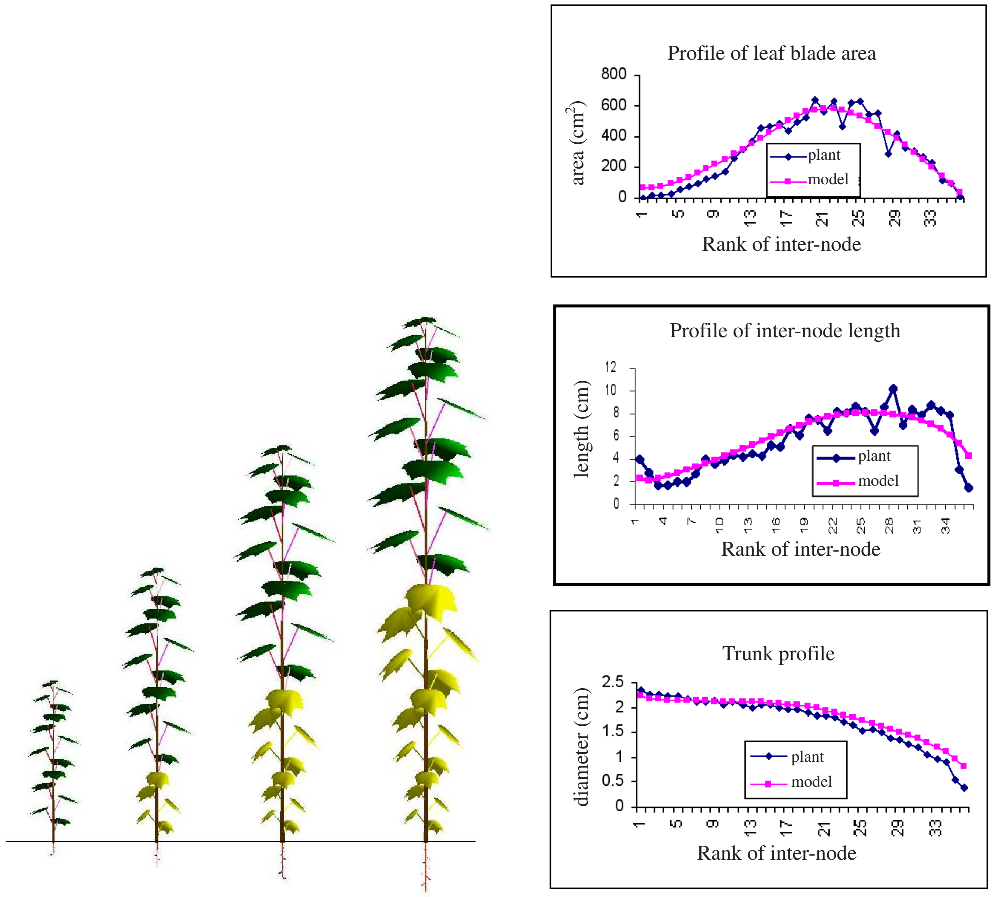


Figure 6: Model calibration for a cotton tree.

topology of the plant [dREF+88]. The following sets of parameters are used to specify a particular plant species, or to aim for a specific behavior or appearance:

- *morphology parameters* defining characteristics such as number of leaves per node, number of flowers per flowering organ, etc.
- *functional parameters* controlling the number of inter-nodes per growth unit or the ramification process.
- *geometrical parameters* describing the shape and placement of organs (phyllotaxy, branching angle, length of inter-nodes etc.)

The cycle begins with the distribution of fresh matter from previous cycles (initially from the seed) to existing and new organs. A global variable represents the amount of fresh matter available,

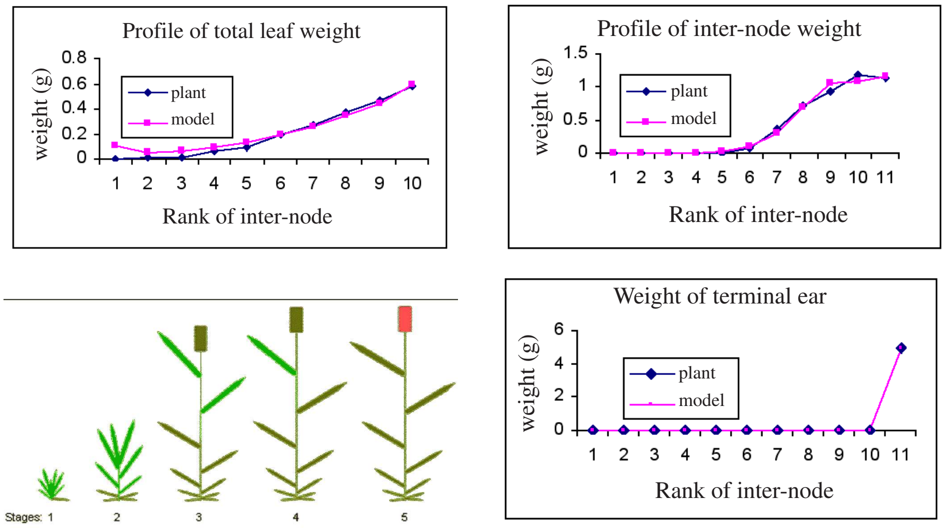


Figure 7: Model calibration for wheat.

and is allocated as explained in Section 3.3.2. Recall that this is done using some of the "old" organs from the previous cycle – trunk rings and fruits– as well as organs that are active from the current cycle. This phase requires knowledge of sink strengths for all organs. These parameters can be obtained for a given species from a calibration experiment as in Section 3.5. Alternatively, typical values can be used (about 0.5 for leaves, 0.08 for inter-nodes, and 0.32 for rings, in arbitrary units). Finally, the dramatic effect of light on the internal development of plants suggests that an explicit retroaction should be used in this phase. That is, the sink strength of each organ can be made to depend on the lighting environment of the organ. We are currently defining the sink strength as a function of the amount of fresh matter created by the organ in the previous cycle. This has the desired effect of slowing down growth in areas which receive little energy, and qualitatively matches observation. Precise validation of this behavior is currently under way but requires very long studies.

The new organs are then set in place on the geometrical structure of the previous cycle following the botanical rules (such as phyllotaxy and branching angle [HOT78]) so as to ensure the 3D growth. Other rules such as competition for space [MP96] can also intervene at this stage. This is done by using a space subdivision structure encoding space occupation. Our system also allows geometric deformations in response to the direction of incident light (photo-tropism), as well as in response to mechanical stress in the branches [FL96].

The production of fresh matter is computed after transforming the plant architecture into an hydraulic structure. In general the simple structure of Section 3.3.1, with independent leaves operating in parallel, is sufficient. But we can actually assemble and solve the complete network of hydric resistances representing the plant.

4 Lighting simulation

This section is entirely devoted to the computation of the light energy distribution inside the plant. As seen in 3.3.1 the production of vegetal matter inside each leaf is conditioned by the amount of incoming radiant energy $\Phi(x,t)$ on the leaf.

To compute $\Phi(x,t)$ we propose a method based on hierarchical radiosity (HR) with clustering and instanciation [SS00]. Hierarchical radiosity with clustering is usually well adapted to treating scenes of various orders of magnitude, thanks to its automated adaptability and to the offered tradeoff between computation time and accuracy. However, such methods are limited in scene size because of the super linear amount of memory they require in terms of the number of input polygons. The main difference between our method and traditional lighting simulation methods is the use of instanciation [SS00]. Instanciation, more commonly used in ray tracing methods, allows to treat arbitrary large scenes by keeping present in memory the only necessary geometry for current calculations. Applying this paradigm to HR with clustering, we thus combine the low memory cost of instanciation ray tracing methods and the stability and controllability of HR with clustering.

Despite its close relationship to HR with clustering, it would be too limiting to simply view our method as an other HR method: because it treats large structures as independent objects exchanging light energy, it can readily be compared to any Monte Carlo approach, where light energy rays between large structures are encapsulated into form factor calculations, rather than randomly dispersed in the model.

4.1 Algorithm overview

The algorithm presented here is inspired from the hierarchical instanciation technique of Soler *et. al* [SS00]. We show that it applies particularly well to the case of vegetation scenes, and focus on the choices made to best adapt the algorithm to specific characteristics of plant models. Let us first shortly review the principle of instanciation for radiosity, refering the reader to [SS00] for technical justifications.

One very eye-striking characteristic of plant models is self-similarity: leaves in a plant are very similar to each other and, up to a large extent, branches look like other branches as well as an entire plant looks like any entire plant of the same species. It is thus possible to approximately represent the geometry of a plant model using a small number of representative elements (branches, leaves, etc) that we can instance in order to build a memory-cheap representation of the plant as shown on figure 8.

Since we will need to know the energy distribution on the real geometry of every part of a tree, the computation can not be performed on a unique geometry shared by the instances. The core idea of the algorithm is to assign macroscopic reflectance and transmittance properties to the representative structures and use them to propagate light inside the model. Then, in order to obtain the distribution of light inside each instance, it is possible to load its real geometry and locally refine the computation. Because plants have self similarity at multiple hierarchical levels (branches, leaves, whole plant), this may include the temporary creation of new instances at lower levels. This is illustrated on figure 9.

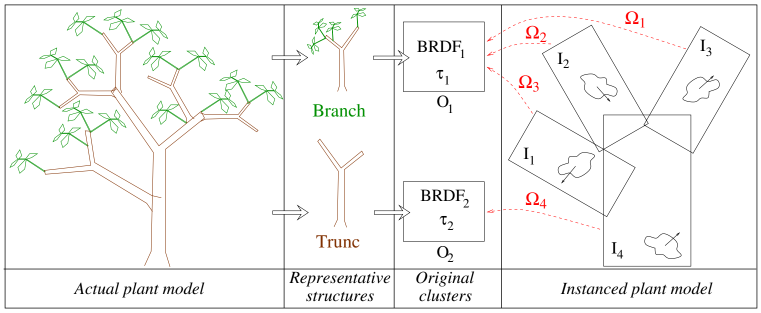


Figure 8: Creation of an instanced representation of a plant model. (1) Each structure of the plant (branch, leaf, trunk) is independently assigned a representative structure, *e.g* a structure from a *plant database* (see section 4.2.2) that sufficiently resembles the current structure. If none is found the current structure is added to the data-base. The transmittance and reflectance properties of representative structures are packed into *original clusters*. (2) the plant is represented as a collection of instances (which are clusters with a specific behavior), each one pointing to the adequate original cluster and equipped with a geometric transformation that permits to perform visibility and reflectance computation on the instance using the information of the original cluster.

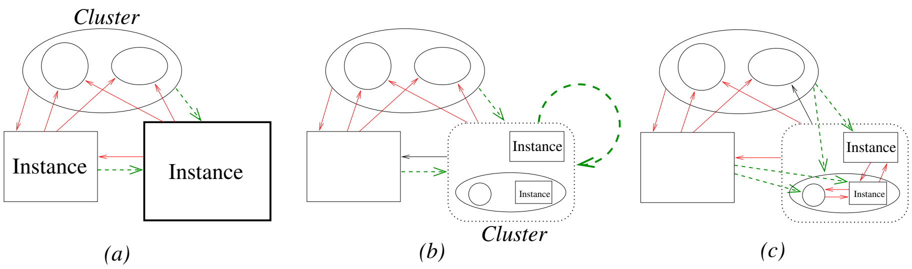
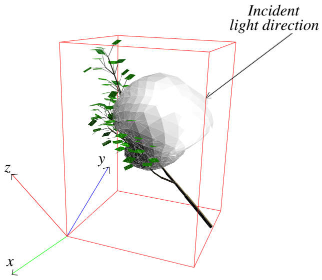


Figure 9: Closer view on the computation of the solution. *left :* a solution as been computed using the reflectance and transmittance properties of instances; links are indicated by arrows. *center :* the bottom left instance is "opened", *e.g* its geometry is loaded into memory. *right :* incoming links (in green) are locally refined as well as the self link that was added on the replacing cluster.

The algorithm eventually reaches portions of the geometry that contains no further instances, in which case it behaves like classical HR with clustering on the local geometry. The local solution is then output to a file. Otherwise, the algorithm is recursively called on all present lower-level instances, after which and the current instance is closed and its solution discarded from memory.

Soler *et. al* showed that the local solution obtained in the current region of the hierarchy is a very good approximation of what the global solution of light energy equilibrium would be at that same place [SS00].

(a) branch, 138 leaves — *(b)* branch, 540 leaves — *(c)* branch, 1506 leaves — *(d)* plant, 2259 leaves — *(e)* plant, 5541 leaves



(f) BRDF sample for (a)

Figure 10: *(a)* to *(e)* Various structures issued from a model of a poplar tree, all represented in their local coordinate system. The number of leaves serves as a discriminant factor for determining which structures can be approximated by an other structure. *(f)* directional value of the BRDF of *(a)* in a given input direction, displayed into the scene coordinate system.

This algorithm is easily controllable: computation time/accuracy trade-off can be readily performed by acting on the refinement threshold of the radiosity links. An other aspect of the controllability is that we can limit the in-depth traversal of the instance hierarchy to a given size of structures and thus obtain a high level solution at a very low computation cost. The accuracy of this solution is much better than that of a clustering algorithm that we would limit to large scale clusters, because of the precise reflectance and transmittance information used at the instance level. This is very interesting if we want to limit interactions with the plant growth simulation engine at large scales: when computing light energy into a forest, the plant growth simulator may be satisfied by the illumination received at the level of branches, that would still be very accurate in that case without going down to the level of leaves.

In the next section we discuss the approximation, computation and storage of light properties of reference structures in a plant, as well as how to choose the representative structures themselves.

4.2 Light properties of plants

4.2.1 Reflectance and transmittance of representative-structures

Representation To represent reflectance functions, we can choose between at least three kinds of functions of various memory costs and accuracy: 1.constant values, *e.g* a single spectrum, like in diffuse radiosity algorithms, 2.mono-directional functions corresponding to the average radiosity of the replaced geometry when stimulated from a given input direction [SS00], and 3.bidirectional functions. Representations 1 and 2 are easily obtained from the BRDF itself.

We represent directional functions as the array of values they take for a set of pre-sampled directions distributed on the unit sphere. These directions are obtained by recursive subdivision of a tetrahedron, giving successive sets of 4, 10,34,130,514... directions associated to the vertices of the resulting polyhedra. Fast access to the values of the functions is provided using linear interpolation on each face of the polyhedra. Any useful information that does not depend on the function values but serves some computation (solid angle attributed to each direction, adjacency relationship between the directions, etc) is shared. Bi-directional functions are also sampled that way in input direction, each direction producing a directional distribution of outgoing light.

The table below sums up the memory cost inherent to various choice for representing reflectance function of a poplar branch of 540 leaves (see figure 10): constant values, mono-directional functions sampled in 130 directions, and bidirectional functions sampled in respectively 34^2,130^2 and 514^2 directions. The first two columns indicates the norm of the difference between the reflectance function and the most accurate 514^2 BRDF. The last column shows the measured memory cost for propagating light in a poplar tree of 5541 leaves (figure 10.e) using 4 levels of instances, and an average of 8 instances and 30 polygons per newly opened instance. 4 original clusters (and thus 4 reflectance functions) were used.

Reflectance function type	Error $\|\cdot\|_\infty$	Error $\|\cdot\|_2$	Function size in kB	Memory in kB
Constant	70 %	15 %	0.012	1
Mono-directional	52 %	11 %	6.168	26
Bi-directional 1	24 %	13 %	13.872	993
Bi-directional 2	11 %	3 %	202.800	1 748
Bi-directional 3	0 %	0 %	3 170.352	13 618

When using bi-directional reflectance functions, unidirectional radiosity and irradiance distributions are necessary for every instance for coherence of the computation. Bidirectional reflectance functions can be quite costly, but these functions are shared by many instances and only a few distinct BRDFs are usually necessary (hardly more than 10) for a given plant.

In the experiment above (which we believe to be representative) the memory cost is essentially driven by the size of the BRDFs we use. Indeed, thanks to the small branching factor (38 in this case) of the hierarchy, the number of instances simultaneously present in memory is small and their cost is overweighted by that of the BRDF for 514^2 directions.

Looking at the associated accuracy, our conclusion is that mono-directional reflectance functions are in average acceptable but still can advantageously be replaced by BRDFs of 130^2 directions.

Just like reflectance functions, transmittance functions of the instances can be represented as using single values, directional 2-D functions or directional 4-D functions. Obviously 4-dimensional transmittance functions are overkill because they would faithfuly represent a geometry that is not the one the instance really replaces, due to the approximate instanciation. Using a single transmittance value [Sil95] (for instance based on the density of clusters) would consider plant structures to be isotropic. This assumption is not valid for a number of structures like plagiotropic branches, for which leaves have a common orientation. We consequently have chosen a directional approximation of the transmittance.

Computation To compute the BRDF of a structure we stimulate the geometry with a light flux of constant intensity and controllable direction, compute the equilibrium of light energy inside the geometry using a HR algorithm and measure the outgoing light intensity and spectrum. Although the sampling itself can be done using graphics hardware, the computation itself is a costly operation that mainly depends on the accuracy threshold of the HR used [Sil95].

Figure 11 gives an example result of such a calculation. It appears clearly that the accuracy threshold must be sufficiently small but not too small to avoid wasting time without any improvement of the BRDF. In practice threshold values of about 20% (relative to the impulse stimulation) provide excellent results. The HR algorithm indeed proceeds an in-depth refinement of light interactions between clusters in the structure. An interaction between two clusters is considered satisfactory when the estimated variance of energy transfers between pairs of polygons of these clusters is smaller than ε.

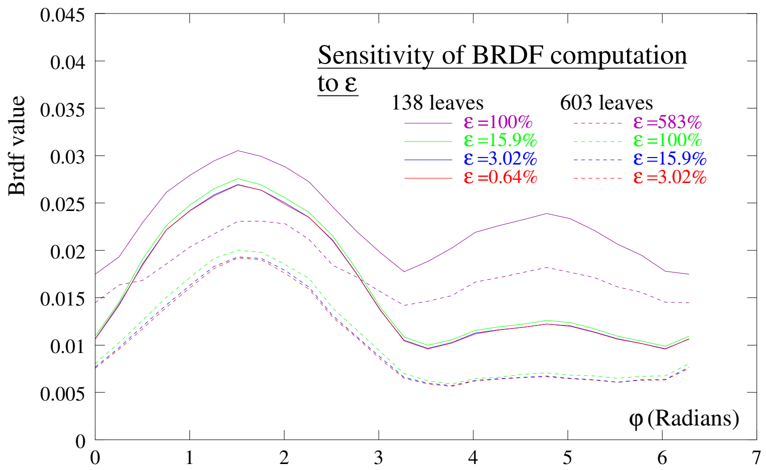


Figure 11: This figure shows cut sections of the BRDFs of two structures (figure 10.a and 10.c computed with a HR algorithm with different accuracy threshold values. For both structures there exists a value of ε under which the BRDFs is nearly indistinguishable from the exact value.

Transmittance functions are computed for each direction as suggested in [SS00], by rendering the corresponding plant structure into an image, that is read to obtain occluded and not occluded pixels.

4.2.2 Instanciation policy

As we saw, the cost of the reflectance function of an instance can be quite large, and the memory cost of the algorithm would be too high if we used a specific BRDF function for each structure of a plant in the scene. On figure 12 we have drawn slices of the BRDF of various structures of a plant and it appears clearly on this example that structures of the same type with similar number of leaves have very similar BRDFs.

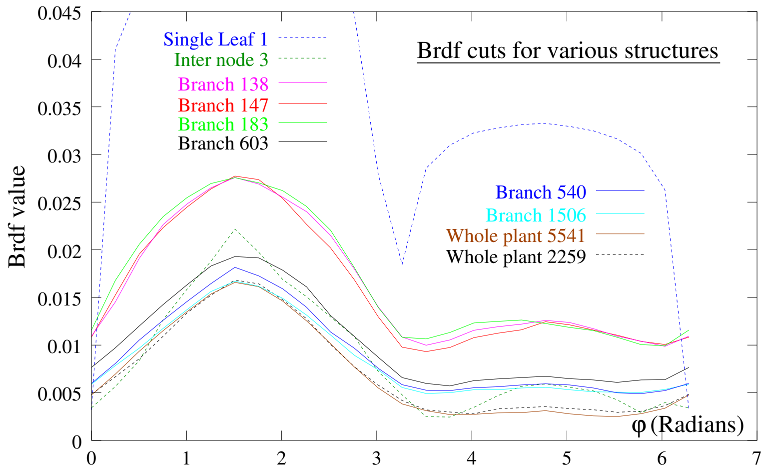


Figure 12: These different curves represent slices of the BRDF of various structures extracted from a model of poplar tree. See figure 10 for an image of branch structures with 138,540 and 1506 leaves and whole plants of 2259 and 5541 leaves. The slices correspond to a lateral input direction of $\theta_{in} = \frac{\pi}{2}$ and $\phi_{in} = \frac{\pi}{2}$ and a lateral output direction of $\theta_{out} = \frac{\pi}{2}$ and varying ϕ_{out}.

This allows us to perform our computation using *approximate instanciation*, *e.g* we only use a small number of representative structures to provide BRDFs for all possible instances. The policy for sharing the BRDFs is for the moment designed by hand for each plant, but it could easily be automated based on the computation of differences between BRDFs of various structures.

Reflectance and transmittance functions for the representative structures are stored in a database. The module that is responsible for choosing which representative structure can approximate a given part of the plant is also in charge of updating on the fly the database, in the case where it can not find a suitable representative structure among already computed ones.

4.2.3 Leaf reflectance and transmittance model

Examining plants, it clearly appears that the reflectance of leaves is not the same on both sides, and is also partially specular mainly on the side facing the sun. According to [BR97] and [Gov95], the transparency of plant leaves is also almost purely diffuse.

For the sake of simplicity, we currently assume diffuse reflection and transmission on leaves. But, although it would requires to store directional distributions of out-coming energy and multi-dimensional reflectance functions, using a directionally-dependent reflectance (or transmittance) model for light transfers on leaves does not cost much more for two reasons : (1) only a small number of leaves are simultaneously present in memory thanks to the instanciation algorithm and thus only a few out-coming light distributions are simultaneously required. (2) the reflectance/transmittance functions of the leaves being identical, they can be shared in memory.

In summary, our model for light exchange at the level of leaves simply consists in transferring the irradiance on each side of a leaf to the radiosity on the other side, after multiplication by a diffuse transmission factor τ. Calling B_i the radiosity of side i of a leaf and I_i its irradiance and R_i its reflectance, we use:

$$B_1 = \rho_1 I_1 + \tau I_2$$
$$B_2 = \rho_2 I_2 + \tau I_1$$

This is computed during the hierarchical push/pull operation [Sil95]. The light energy pulled up in the hierarchy is the average value of the energies on both sides of the leaf.

4.3 Implementation issues

4.3.1 Light sources

Our system currently handles directional light sources, *e.g* that illuminate all objects from the same direction (thus producing hard shadows), infinite light sources, *e.g* sources that are composed of an independent set of incoming directions packed into a fixed solid angle scope (thus producing soft shadows), and local diffuse light sources usual to standard radiosity packages.

In order to simulate a sky dome illumination during a long period (which is the more realistic situation for a tree) we combine an infinite light source covering the sky hemisphere and a number of infinite light sources with small solid angles to represent sampled positions of the sun during the period.

Radiosity-like diffuse light sources are not common in nature but still can be used to simulate the illumination due to the environment.

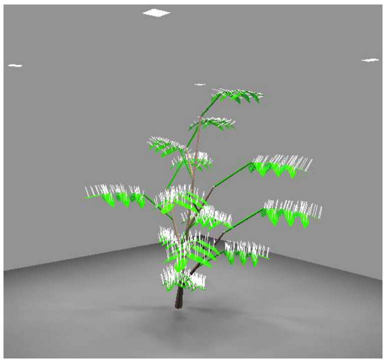
4.3.2 Collecting information for the growth simulator

The physiological plant growth simulator needs to know for each growth cycle the intensity of light energy received by the leaves as well as the average direction of incoming light. Both quantities are obtained on the fly during the computation, when the concerned geometry is present in memory: the irradiance is obtained by summing the contribution of the links that end on all parents of the current leaf. The main irradiance direction v_i for leaf i is obtained by summing the irradiance vector contributions $I(E_j)$ from the emitters E_j of the links on the parents of i in the following way :

$$\vec{v}_i = \frac{1}{\sum_{E_j \in P(i)} I(E_j)} \sum_{E_j \in P(i)} I(E_j)\vec{v}(E_j)$$

with

$$\vec{v}(E_j) = \int_{E_j} \frac{y-x}{||y-x||^3} \cos\theta dy$$



Using this formula, the more "directional" the incoming light is, the larger is the norm of $\vec{v}_i$. If incoming light is uniformly distributed in upcoming directions around a leaf, $\vec{v}_i$ is $\vec{0}$. The image above shows these vectors for each polygon of the leaves of a simple plant lit by four light sources.

5 Results and examples

Influence of instanciation The use of instanciation to facilitate the computation of the distribution of light energy inside the plants adds approximations to the light energy distribution. We therefore measured the impact of these approximations on the growth simulation as compared to what is obtained using standard HR with clustering.

Because the plant growth simulator integrates the light received by a particular region of the plant over a certain period of time and of space, local variations of the solution in time and space may not affect the result of the simulation, i.e the shape of the plant. We will thus directly measure the impact of these approximations on the plant model itself.

It is not easy to measure the "difference" between two plants grown under slightly different conditions because their topology may not be the same. To cope with this, we have desactivated the fall of dead structures and chosen a plant for which the branching factor is constant. In that case the correspondance between equivalent structures in two plants of the same age becomes easy.

Figure 13 shows the error associated with the use of instanciation when computing the light distribution during a plant growth simulation, as well as computation times and memory usage. This example can also serve as a reference for computation times and memory costs of our method.

The conclusion for this small experiment is that instanciation does perturb the simulation, but in a rather acceptable manner. Limiting the depth of the computation to instances of at least 10 only brings a small gain in computation time and saves the memory cost of light properties of small structures (*e.g* two BRDFs in that case.

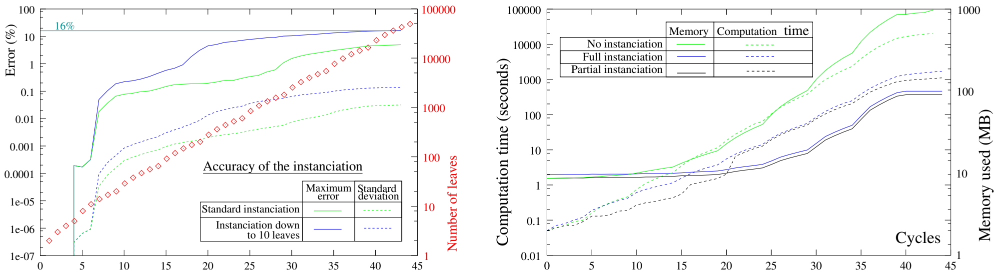


Figure 13: *left :* Maximum value and standard deviation of the distance between structures of a plant grown using instanciation and using standard HR with clustering. In blue: instanciation was limited to structures of more than 10 leaves, after which a simple gather and pushpull was performed on the structures, thus inducing more error. In green: standard instanciation. *Right:* The computation times and memory cost corresponding to each growth cycle are displayed for these two experiments, as well as for the growth without instanciation. *Bottom left :* view of test scene at cycle 37. Note that when using instanciation, lighting simulation typically takes a fixed proportion of total growth simulation computation time (roughly 80%)

Indirect lighting We present a test on indirect lighting to determine the importance of simulating this phenomenon during the growth process. This justifies the use of an algorithm capable of such a simulation rather than simply measuring direct lighting through the vegetation.

Within the visible light spectrum range, diffusion of light inside plants foliage is limited by the small reflectance and transmittance values of the leaves (about 0.04: see [Gov95] for typical examples). Figure 14 shows the same plant grown with direct lighting only on the left, and normal lighting on the right. Measuring the difference between the two results using the norm described above, we find (and it is visible on the pictures) that the error is more than 10% of the size of the plant (for comparison, the error due to instanciation approximations is less than 0.1% at that cycle).

In the IR domain, internal diffusion of light becomes even more significant because reflectances and transmittances of plant leaves take values up to 40% [Gov95]. Simulating the IR distribution is useful to internal temperature profile in plants.

Indirect lighting may not only come from the vegetation itself but also from objects in the scene. A rather common example is that of a plant growing next to a wall (in front of a house for instance). Then a significant part of the light energy received by the plant comes from the surrounding objects.

Various examples (also see video) Figure 15 (*top*) shows a plant that starts its growth behind a wall. The production of vegetal matter and the sink of organs that receive light, as well as the seek for incoming light, favore the growth of vegetative axis trough the hole. Successive images correspond

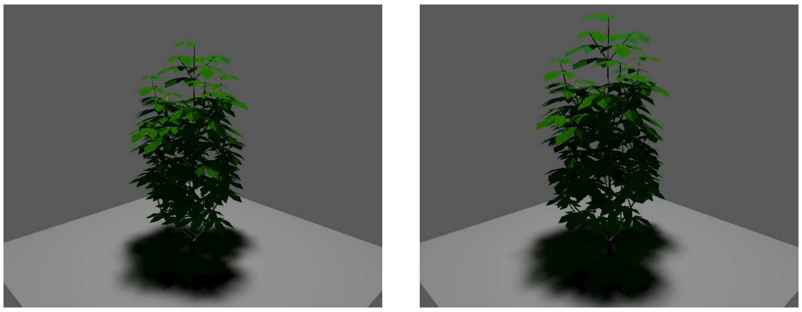


Figure 14: Compared effect of accounting or not for indirect lighting.

to cycles 2, 9, 13,17 and 25. These images were rendered in a second pass, using our radiosity simulator, from the geometric data produced during the simulation by the plant growth engine, but using a set of parameters that is more specifically adapted to visual results. Note that such an accuracy was not at all necessary for growing the plant itself. Total simulation time for growing this plant is less that 20 mins on a workstation equipped with a 250 MHz $R12000$ Processor.

On figure 15 (*middle*) we show an example of a plant growing under a light source. Images correspond to cycles 4,8,15 and 37. It appears clearly at cycle 8 that the overall balance of the plant is influenced by light. Indeed the part that is closer to the light source receives more energy and thus produces more vegetal matter, hence the higher density of the foliage. Directionality of the axis toward the light source is not as obvious here as it was on the previous exemple (although it can be seen on the second image) because of the larger rigidity of the branches for that species. The third image shows the transparency effect of the leaves quite well. Total simulation time here is 1 hour 30 minutes.

The last row of Figure 15 shows four plants grown under a sun light source in competition for light and space. The rendering was done with OpenGL using a hand made tool to interpolate the geometry between cycles. This has permitted to visualy simulate the fall of the leaves and show the trunks.

6 Conclusions

We have presented a complete system for physiological plant growth simulation, based on the accurate computation of light energy distribution inside the plants. The growth engine itself is based on a detailed simulation of the internal functioning of the plant, accounting for incoming light energy, and has received very good calibrations on real plants data. We have described how to accurately compute the light energy distribution in the plants using a performant hierarchical radiosity algorithm with instanciation.

Our system has applications in numerous fields : compute geometric plant models adapted to their environment for computer graphics applications, simulate plantation growth under various conditions for optimization of cultures in agronomy or for urban modeling and site planning.

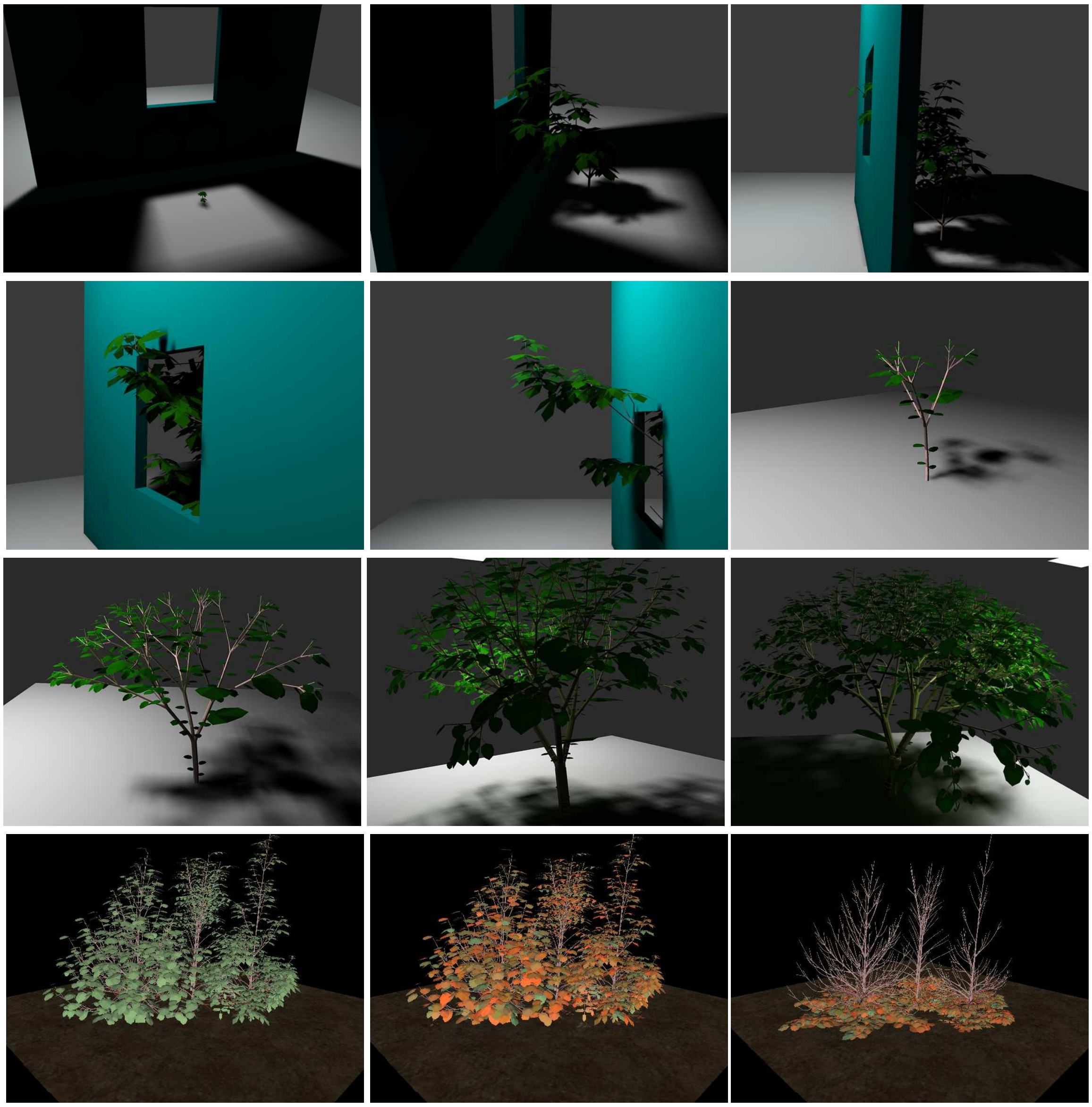


Figure 15: Various examples. See text. *First serie :* Example of a plant that automatically grows towards the light source through a hole in a wall. *Second serie :* four stages in the growth of a tree influenced by a local light source. *Third serie :* four plants that compete for space and light.

Future works include the calibration of the reaction of the growth simulator to lighting simulation: the growth engine itself is perfectly calibrated to statistical data measured on real plants concerning vegetal matter production and organ sizes. However we still need to infer the parameters involved into the reactions to light. This can be done using the same hidden parameters identification techniques we used for determining sink strength, expansion functions and hydric resistances. This calibration needs a complex experiment setup to provide constant light conditions and proper measure of incident light energy on leaves and geometric plant characteristics, on a sufficient number of plants to be able to obtains relevant statistics. We are currently working on such experiments and will be able to provide the actual values of these parameters for various kinds of plants.

The interaction between the plant growth engine and the lighting simulation algorithm takes place, up to now, at the level of the leaves. As we explained it would be more judicious to limit the computation of incident light at the level of larger structures when the number of leaves becomes too large, but still keeping an accurate solution. We saw that this facility is already provided by our lighting simulation algorithm but this still needs to check the sensitivity of the growth simulator to local averaging of incoming light inside large structures. This opens the way to accurately simulating the growth of coniferous trees, for instance.

In our introduction we have already insisted on the fact that the geometric dressing of the plant models produced by the growth simulator is not our main goal. This task indeed has already been covered widely in recent works and is not required for proper simulation. A number of existing algorithms can however be used to produce nice looking images from the topologic structure of the plant.

References

[BBJ+98] F. Blaise, J.F. Barczi, M. Jaeger, P. Dinouard, and Ph. de Reffye. Simulation of the growth of plants, modeling of metamorphosis and spatial interactions in the architecture and development of plants. *Cyberworlds*, 6:81–109, 1998.

[BGP91] Christoph C. Borel, Siegfried A. W. Gerstl, and Bill J. Powers. The Radiosity Method in Optical Remote Sensing of Structured 3-D Surfaces. *Remote Sensing of the Environment*, 36:13–44, May 1991.

[BR97] G. V. G. Baranoski and J. G. Rokne. An algorithmic reflectance and transmittance model for plant tissue. In *Computer Graphics Forum (Proc. Eurographics '97)*, volume 16(3), August 1997.

[CAB98] Michael Chelle, Bruno Andrieu, and Kadi Bouatouch. Nested radiosity for plant canopies. *The Visual Computer*, 14(3):109–125, 1998.

[DM87] J. Dauzat and Eroy M.N. Simulating light regime and intercrop yields in coconut based farming systems. *European Journal of Agronomy*, 7:63–74, 1987.

[dRBC+99] Ph. (de) Reffye, F. Blaise, S. Chemouny, S. Jaffuel, Th. Fourcaud, and F. Houllier. Calibration of a hydraulic architecture-based growth model of cotton plants. *Agronomie*, 19:265–280, 1999.

[dREC91] P. (de) Reffye, E. Elguero, and E. Costes. Growth units construction in trees: a stochastic approach. In *9th Seminar of the Theoritical Biology School (Solignac)*, volume 39(3-4), pages 325–342. Acta Biotheoritica, September 1991.

[dREF+88] Ph. de Reffye, C. Edelin, J. Françon, M. Jaeger, and C. Puech. Plant models faithful to botanical structure and development. In *Computer Graphics(Proceedings of SIGGRAPH 88)*, volume 22(4), pages 151–158, 1988.

[dRFB+96] Ph. (de) Reffye, Th. Fourcaud, F. Blaise, D. Barthélémy, and F. Houllier. An ecophysiological model for tree growth and tree architecture. In *Workshop on Functional Structural Tree Models. Helsinki*. Silva Fennica eds., 1996.

[dRH97] Ph. (de) Reffye and F. Houllier. Modelling plant growth and architecture: some recent advances and applications to agronomy and forestery. *Curr.Sci.*, 73:984–992, 1997.

[FL96] T. Fourcaud and P. Lac. Mechanical analysis of the form and internal stresses of a growing tree by the finite element method. In A.E. Engin, editor, *ESDA proceedings*, volume 5, bioengineering, pages 213–220. Inra/Elsevier, Paris, 1996.

[GGCC97] C. Godin, Y. Guedon, E. Costes, and Y Caraglio. Mesuring and analysing plants with the amapmod software. *Plants to Ecosystems:Advances in Computational Life Sciences*, 1:63–94, 1997.

[Gov95] Y. M. Govaerts. *A Model of Light Scattering in Three-Dimensional Plant Canopies: A Monte Carlo Ray Tracing Approach*. PhD thesis, Departement de Physique, Universitat Catholique de Louvain, Louvain, Belgium, 1995.

[Gre89] Ned Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 175–184, July 1989.

[GRT91] N. S. Goel, I. Rozehnal, and R. L. Thompson. A computer graphics based model for scattering from objects of arbitrary shapes in the optical region. *Remote Sensing of Environment*, 36(2):73–104, 1991.

[Han95] J.S. Hanan. Virtual plants–integrating architectural and physiological models. In *Proceedings of Int. Congr. Modelling and Simulation*, 1995.

[HOT78] F. Hallé, R.A.A Oldeman, and P.B Tomlinson. *Tropical Trees and Forests*. Springer Verlag, Berlin, Heidelberg, New-York, 1978.

[JA88] Ross J.K. and Marshak A.L. Calculation of canopy bidirectional reflectance using the monte carlo method. *Remote Sensing of the Environment*, 24:213–225, 1988.

[Jal98] E. Jallas. *Improved model-based decision support by modeling cotton variability and using evolutionary algorithms*. PhD thesis, Mississipi State University, 1998.

[Kur94] W. Kurth. Morphological models of plant growth : possibilities and ecological relevance. *Ecological modelling*, 75-76:299–308, 1994.

[MMKW97] Nelson Max, Curtis Mobley, Brett Keating, and En-Hua Wu. Plane-parallel radiance transport for global illumination in vegetation. In Julie Dorsey and Phillip Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 239–250, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.

[MP96] Radomír Měch and Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 397–410. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[PHM93] P. Prusinkiewicz, M. Hammel, and E. Mjolsness. Animation of plant development. In *Proceedings of SIGGRAPH '93 (Anaheim, California)*, pages 351–360. ACM SIGGRAPH, New York, August 1993.

[PLH88] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan. Developmental models of herbaceous plants for computer imagery purposes. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, volume 22(4), pages 141–150, 1988.

[PSN+96] J. Perttunen, R. Sievänen, E. Nikinmaa, H. Salminen, H. Saarenmaa, and J. Väkevä. Lignum: a tree model based on simple structural units. *Annals of Botany*, 77:87–98, 1996.

[Sil95] François Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3), September 1995. (a preliminary version appeared in the fifth Eurographics workshop on rendering, Darmstadt, Germany, June 1994).

[SS00] Cyril Soler and François Sillion. Hierarchical instantiation for radiosity. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques '00*, pages 173–184, New York, NY, 2000. Springer Wien.

[WP95] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 119–128. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399