

Visual Models of Plants Interacting with Their Environment

Radomír Měch and Przemysław Prusinkiewicz¹

University of Calgary

ABSTRACT

Interaction with the environment is a key factor affecting the development of plants and plant ecosystems. In this paper we introduce a modeling framework that makes it possible to simulate and visualize a wide range of interactions at the level of plant architecture. This framework extends the formalism of Lindenmayer systems with constructs needed to model bi-directional information exchange between plants and their environment. We illustrate the proposed framework with models and simulations that capture the development of tree branches limited by collisions, the colonizing growth of clonal plants competing for space in favorable areas, the interaction between roots competing for water in the soil, and the competition within and between trees for access to light. Computer animation and visualization techniques make it possible to better understand the modeled processes and lead to realistic images of plants within their environmental context.

CR categories: F.4.2 [Mathematical Logic and Formal Languages]: Grammars and Other Rewriting Systems: *Parallel rewriting systems*, I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, I.6.3 [Simulation and Modeling]: Applications, J.3 [Life and Medical Sciences]: Biology.

Keywords: scientific visualization, realistic image synthesis, software design, L-system, modeling, simulation, ecosystem, plant development, clonal plant, root, tree.

1 INTRODUCTION

Computer modeling and visualization of plant development can be traced back to 1962, when Ulam applied cellular automata to simulate the development of branching patterns, thought of as an abstract representation of plants [53]. Subsequently, Cohen presented a more realistic model operating in continuous space [13], Linden-

¹Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4 (mech|pwp@cpsc.ucalgary.ca)

mayer proposed the formalism of L-systems as a general framework for plant modeling [38, 39], and Honda introduced the first computer model of tree structures [32]. From these origins, plant modeling emerged as a vibrant area of interdisciplinary research, attracting the efforts of biologists, applied plant scientists, mathematicians, and computer scientists. Computer graphics, in particular, contributed a wide range of models and methods for synthesizing images of plants. See [18, 48, 54] for recent reviews of the main results.

One aspect of plant structure and behavior neglected by most models is the interaction between plants and their environment (including other plants). Indeed, the incorporation of interactions has been identified as one of the main outstanding problems in the domain of plant modeling [48] (see also [15, 18, 50]). Its solution is needed to construct predictive models suitable for applications ranging from computer-assisted landscape and garden design to the determination of crop and lumber yields in agriculture and forestry.

Using the information flow between a plant and its environment as the classification key, we can distinguish three forms of interaction and the associated models of plant-environment systems devised to date:

1. The plant is affected by global properties of the environment, such as day length controlling the initiation of flowering [23] and daily minimum and maximum temperatures modulating the growth rate [28].
2. The plant is affected by local properties of the environment, such as the presence of obstacles controlling the spread of grass [2] and directing the growth of tree roots [26], geometry of support for climbing plants [2, 25], soil resistance and temperature in various soil layers [16], and predefined geometry of surfaces to which plant branches are pruned [45].
3. The plant interacts with the environment in an information feedback loop, where the environment affects the plant and the plant reciprocally affects the environment. This type of interaction is related to *sighted* [4] or *exogenous* [42] mechanisms controlling plant development, in which parts of a plant influence the development of other parts of the same or a different plant through the space in which they grow. Specific models capture:
 - competition for *space* (including collision detection and access to light) between segments of essentially two-dimensional schematic branching structures [4, 13, 21, 22, 33, 34, 36];
 - competition between root tips for *nutrients* and *water* transported in soil [12, 37] (this mechanism is related to competition between growing branches of corals and sponges for nutrients diffusing in water [34]);

- competition for *light* between three-dimensional shoots of herbaceous plants [25] and branches of trees [9, 10, 11, 15, 33, 35, 52].

Models of exogenous phenomena require a comprehensive representation of both the developing plant and the environment. Consequently, they are the most difficult to formulate, implement, and document. Programs addressed to the biological audience are often limited to narrow groups of plants (for example, poplars [9] or trees in the pine family [21]), and present the results in a rudimentary graphical form. On the other hand, models addressed to the computer graphics audience use more advanced techniques for realistic image synthesis, but put little emphasis on the faithful reproduction of physiological mechanisms characteristic to specific plants.

In this paper we propose a general *framework* (defined as a modeling methodology supported by appropriate software) for modeling, simulating, and visualizing the development of plants that bi-directionally interact with their environment. The usefulness of modeling frameworks for simulation studies of models with complex (emergent) behavior is manifested by previous work in theoretical biology, artificial life, and computer graphics. Examples include cellular automata [51], systems for simulating behavior of cellular structures in discrete [1] and continuous [20] spaces, and L-system-based frameworks for modeling plants [36, 46]. Frameworks may have the form of a general-purpose simulation program that accepts models described in a suitable mini-language as input, *e.g.* [36, 46], or a set of library programs [27]. Compared to special-purpose programs, they offer the following benefits:

- At the conceptual level, they facilitate the design, specification, documentation, and comparison of models.
- At the level of model implementation, they make it possible to develop software that can be reused in various models. Specifically, graphical capabilities needed to visualize the models become a part of the modeling framework, and do not have to be reimplemented.
- Finally, flexible conceptual and software frameworks facilitate interactive experimentation with the models [46, Appendix A].

Our framework is intended both for purpose of image synthesis and as a research and visualization tool for model studies in plant morphogenesis and ecology. These goals are addressed at the levels of the simulation system and the modeling language design. The underlying paradigm of plant-environment interaction is described in Section 2. The resulting design of the simulation software is outlined in Section 3. The language for specifying plant models is presented in Section 4. It extends the concept of environmentally-sensitive L-systems [45] with constructs for bi-directional communication with the environment. The following sections illustrate the proposed framework with concrete models of plants interacting with their environment. The examples include: the development of planar branching systems controlled by the crowding of apices (Section 5), the development of clonal plants controlled by both the crowding of ramets and the quality of terrain (Section 6), the development of roots controlled by the concentration of water transported in the soil (Section 7), and the development of tree crowns affected by the local distribution of light (Section 8) The paper concludes with an evaluation of the results and a list of open problems (Section 9).

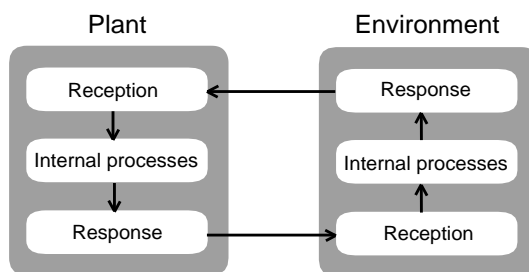


Figure 1: Conceptual model of plant and environment treated as communicating concurrent processes

2 CONCEPTUAL MODEL

As described by Hart [30], every environmentally controlled phenomenon can be considered as a chain of causally linked events. After a stimulus is perceived by the plant, information in some form is transported through the plant body (unless the site of stimulus perception coincides with the site of response), and the plant reacts. This reaction reciprocally affects the environment, causing its modification that in turn affects the plant. For example, roots growing in the soil can absorb or extract water (depending on the water concentration in their vicinity). This initiates a flow of water in the soil towards the depleted areas, which in turn affects further growth of the roots [12, 24].

According to this description, the interaction of a plant with the environment can be conceptualized as two concurrent processes that communicate with each other, thus forming a feedback loop of information flow (Figure 1). The plant process performs the following functions:

- reception of information about the environment in the form of scalar or vector values representing the stimuli perceived by specific organs;
- transport and processing of information inside the plant;
- generation of the response in the form of growth changes (*e.g.* development of new branches) and direct output of information to the environment (*e.g.* uptake and excretion of substances by a root tip).

Similarly, the environmental process includes mechanisms for the:

- perception of the plant's actions;
- simulation of internal processes in the environment (*e.g.* the diffusion of substances or propagation of light);
- presentation of the modified environment in a form perceivable by the plant.

The design of a simulation system based on this conceptual model is presented next.

3 SYSTEM DESIGN

The goal is to create a framework, in which a wide range of plant structures and environments can be easily created, modified, and

used for experimentation. This requirement led us to the following design decisions:

- The plant and the environment should be modeled by separate programs and run as two communicating processes. This design is:
 - compatible with the assumed conceptual model of plant-environment interaction (Figure 1);
 - consistent with the principles of structured design (modules with clearly specified functions jointly contribute to the solution of a problem by communicating through a well defined interface; information local to each module is hidden from other modules);
 - appropriate for interactive experimentation with the models; in particular, changes in the plant program can be implemented without affecting the environmental program, and *vice versa*;
 - extensible to distributed computing environments, where different components of a large ecosystem may be simulated using separate computers.
- The user should have control over the type and amount of information exchanged between the processes representing the plant and the environment, so that all the needed but no superfluous information is transferred.
- Plant models should be specified in a language based on L-systems, equipped with constructs for bi-directional communication between the plant and the environment. This decision has the following rationale:
 - A succinct description of the models in an interpreted language facilitates experimentation involving modifications to the models;
 - L-systems capture two fundamental mechanisms that control development, namely flow of information from a mother module to its offspring (cellular descent) and flow of information between coexisting modules (endogenous interaction) [38]. The latter mechanism plays an essential role in transmitting information from the site of stimulus perception to the site of the response. Moreover, L-systems have been extended to allow for input of information from the environment (see Section 4);
 - Modeling of plants using L-systems has reached a relatively advanced state, manifested by models ranging from algae to herbaceous plants and trees [43, 46].
- Given the variety of processes that may take place in the environment, they should be modeled using special-purpose programs.
- Generic aspects of modeling, not specific to particular models, should be supported by the modeling system. This includes:
 - an L-system-based plant modeling program, which interprets L-systems supplied as its input and visualizes the results, and
 - the support for communication and synchronization of processes simulating the modeled plant and the environment.

A system architecture stemming from this design is shown in Figure 2. We will describe it from the perspective of extensions to the formalism of L-systems.

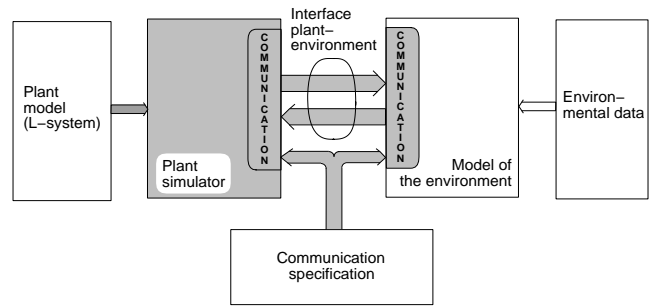


Figure 2: Organization of the software for modeling plants interacting with their environment. Shaded rectangles indicate components of the modeling framework, clear rectangles indicate programs and data that must be created by a user specifying a new model of a plant or environment. Shaded arrows indicate information exchanged in a standardized format.

4 OPEN L-SYSTEMS

Historically, L-systems were conceived as closed cybernetic systems, incapable of simulating any form of communication between the modeled plant and its environment. In the first step towards the inclusion of environmental factors, Rozenberg defined *table L-systems*, which allow for a change in the set of developmental rules (the production set of the L-system) in response to a change in the environment [31, 49]. Table L-systems were applied, for example, to capture the switch from the production of leaves to the production of flowers by the apex of a plant due to a change in day length [23]. *Parametric L-systems* [29, 46], introduced later, made it possible to implement a variant of this technique, with the environment affecting the model in a quantitative rather than qualitative manner. In a case study illustrating this possibility, weather data containing daily minimum and maximum temperatures were used to control the rate of growth in a bean model [28]. *Environmentally-sensitive L-systems* [45] represented the next step in the inclusion of environmental factors, in which local rather than global properties of the environment affected the model. The new concept was the introduction of query symbols, returning current position or orientation of the turtle in the underlying coordinate system. These parameters could be passed as arguments to user-defined functions, returning local properties of the environment at the queried location. Environmentally-sensitive L-systems were illustrated by models of topiary scenes. The environmental functions defined geometric shapes, to which trees were pruned.

Open L-systems, introduced in this paper, augment the functionality of environmentally-sensitive L-systems using a reserved symbol for bilateral communication with the environment. In short, parameters associated with an occurrence of the communication symbol can be set by the environment and transferred to the plant model, or set by the plant model and transferred to the environment. The environment is no longer represented by a simple function, but becomes an active process that may react to the information from the plant. Thus, plants are modeled as open cybernetic systems, sending information to and receiving information from the environment.

In order to describe open L-systems in more detail, we need to recall the rudiments of L-systems with turtle interpretation. Our presentation is reproduced from [45].

An L-system is a parallel rewriting system operating on branching structures represented as *bracketed strings* of symbols with associated numerical parameters, called *modules*. Matching pairs of square brackets enclose branches. Simulation begins with an initial string called the *axiom*, and proceeds in a sequence of discrete *derivation steps*. In each step, *rewriting rules* or *productions* replace all modules in the predecessor string by successor modules. The applicability of a production depends on a predecessor's context (in context-sensitive L-systems), values of parameters (in productions guarded by conditions), and on random factors (in stochastic L-systems). Typically, a production has the format:

$$id : lc < pred > rc : cond \rightarrow succ : prob$$

where *id* is the production identifier (label), *lc*, *pred*, and *rc* are the left context, the strict predecessor, and the right context, *cond* is the condition, *succ* is the successor, and *prob* is the probability of production application. The strict predecessor and the successor are the only mandatory fields. For example, the L-system given below consists of axiom ω and three productions p_1 , p_2 , and p_3 .

$$\begin{aligned} \omega &: A(1)B(3)A(5) \\ p_1 &: A(x) \rightarrow A(x+1) : 0.4 \\ p_2 &: A(x) \rightarrow B(x-1) : 0.6 \\ p_3 &: A(x) < B(y) > A(z) : y < 4 \rightarrow B(x+z)[A(y)] \end{aligned}$$

The stochastic productions p_1 and p_2 replace module $A(x)$ by either $A(x+1)$ or $B(x-1)$, with probabilities equal to 0.4 and 0.6, respectively. The context-sensitive production p_3 replaces a module $B(y)$ with left context $A(x)$ and right context $A(z)$ by module $B(x+z)$ supporting branch $A(y)$. The application of this production is guarded by condition $y < 4$. Consequently, the first derivation step may have the form:

$$A(1)B(3)A(5) \Rightarrow A(2)B(6)[A(3)]B(4)$$

It was assumed that, as a result of random choice, production p_1 was applied to the module $A(1)$, and production p_2 to the module $A(5)$. Production p_3 was applied to the module $B(3)$, because it occurred with the required left and right context, and the condition $3 < 4$ was true.

In the L-systems presented as examples we also use several additional constructs (*cf.* [29, 44]):

- Productions may include statements assigning values to local variables. These statements are enclosed in curly braces and separated by semicolons.
- The L-systems may also include arrays. References to array elements follow the syntax of C; for example, $MaxLen[order]$.
- The list of productions is ordered. In the deterministic case, the first matching production applies. In the stochastic case, the set of all matching productions is established, and one of them is chosen according to the specified probabilities.

For details of the L-system syntax see [29, 43, 46].

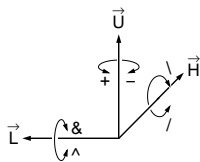


Figure 3: Controlling the turtle in three dimensions

In contrast to the parallel application of productions in each derivation step, the interpretation of the resulting strings proceeds sequentially, with reserved modules acting as commands to a LOGO-style turtle [46]. At any point of the string, the *turtle state* is characterized by a position vector \vec{P} and

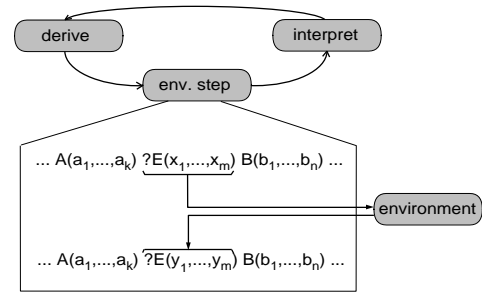


Figure 4: Information flow during the simulation of a plant interacting with the environment, implemented using an open L-system

three mutually perpendicular orientation vectors \vec{H} , \vec{U} , and \vec{L} , indicating the turtle's heading, the up direction, and the direction to the left (Figure 3). Module F causes the turtle to draw a line in the current direction. Modules $+$, $-$, $\&$, \wedge , $/$ and \backslash rotate the turtle around one of the vectors \vec{H} , \vec{U} , or \vec{L} , as shown in Figure 3. The length of the line and the magnitude of the rotation angle can be given globally or specified as parameters of individual modules. During the interpretation of branches, the opening square bracket pushes the current position and orientation of the turtle on a stack, and the closing bracket restores the turtle to the position and orientation popped from the stack. A special interpretation is reserved for the module $\%$, which cuts a branch by erasing all symbols in the string from the point of its occurrence to the end of the branch [29]. The meaning of many symbols depends on the context in which they occur; for example, $+$ and $-$ denote arithmetic operators as well as modules that rotate the turtle.

The turtle interpretation of L-systems described above was designed to visualize models in a postprocessing step, with no effect on the L-system operation. Position and orientation of the turtle are important, however, while considering environmental phenomena, such as collisions with obstacles and exposure to light. The environmentally-sensitive extension of L-systems makes these attributes accessible during the rewriting process [45]. The generated string is interpreted after each derivation step, and turtle attributes found during the interpretation are returned as parameters to reserved *query modules*. Syntactically, the query modules have the form $?X(x, y, z)$, where $X=P, H, U$, or L . Depending on the actual symbol X , the values of parameters x , y , and z represent a position or an orientation vector.

Open L-systems are a generalization of this concept. *Communication modules* of the form $?E(x_1, \dots, x_m)$ are used both to send and receive environmental information represented by the values of parameters x_1, \dots, x_m (Figure 4). To this end, the string resulting from a derivation step is scanned from left to right to determine the state of the turtle associated with each symbol. This phase is similar to the graphical interpretation of the string, except that the results need not be visualized. Upon encountering a communication symbol, the plant process creates and sends a message to the environment including all or a part of the following information:

- the address (position in the string) of the communication module (mandatory field needed to identify this module when a reply comes from the environment),
- values of parameters x_i ,
- the state of the turtle (coordinates of the position and orientation

vector, as well as some other attributes, such as current line width),

- the type and parameters of the module following the communication module in the string (not used in the examples discussed in this paper).

The exact message format is defined in a *communication specification file*, shared between the programs modeling the plant and the environment (Figure 2). Consequently, it is possible to include only the information needed in a particular model in the messages sent to the environment. Transfer of the last message corresponding to the current scan of the string is signaled by a reserved end-of-transmission message, which may be used by the environmental process to start its operation.

The messages output by the plant modeling program are transferred to the process that simulates the environment using an interprocess communication mechanism provided by the underlying operating system (a pair of UNIX pipes or shared memory with access synchronized using semaphores, for example). The environment processes that information and returns the results to the plant model using messages in the following format:

- the address of the target communication module,
- values of parameters y_i carrying the output from the environment.

The plant process uses the received information to set parameter values in the communication modules (Figure 4). The use of addresses makes it possible to send replies only to selected communication modules. Details of the mapping of messages received by the plant process to the parameters of the communication modules are defined in the communication specification file.

After all replies generated by the environment have been received (a fact indicated by an end-of-transmission message sent by the environment), the resulting string may be interpreted and visualized, and the next derivation step may be performed, initiating another cycle of the simulation.

Note that, by preceding every symbol in the string with a communication module it is possible to pass complete information about the model to the environment. Usually, however, only partial information about the state of a plant is needed as input to the environment. Proper placement of communication modules in the model, combined with careful selection of the information to be exchanged, provide a means for keeping the amount of transferred information at a manageable level.

We will illustrate the operation of open L-systems within the context of complete models of plant-environment interactions, using examples motivated by actual biological problems.

5 A MODEL OF BRANCH TIERS

Background. Apical meristems, located at the endpoints of branches, are engines of plant development. The apices grow, contributing to the elongation of branch segments, and from time to time divide, spawning the development of new branches. If all apices divided periodically, the number of apices and branch segments would increase exponentially. Observations of real branching structures show, however, that the increase in the number of segments is less than exponential [8]. Honda and his collaborators modeled several hypothetical mechanisms that may control the extent of

branching in order to prevent overcrowding [7, 33] (see also [4]). One of the models [33], supported by measurements and earlier simulations of the tropical tree *Terminalia catappa* [19], assumes an exogenous interaction mechanism. *Terminalia* branches form horizontal tiers, and the model is limited to a single tier, treated as a two-dimensional structure. In this case, the competition for light effectively amounts to collision detection between the apices and leaf clusters. We reproduce this model as the simplest example illustrating the methodology proposed in this paper.

Communication specification. The plant communicates with the environment using communication modules of the form $?E(x)$. Messages sent to the environment include the turtle position and the value of parameter x , interpreted as the vigor of the corresponding apex. On this basis, the environmental process determines the fate of each apex. A parameter value of $x=0$ returned to the plant indicates that the development of the corresponding branch will be terminated. A value of $x=1$ allows for further branching.

The model of the environment. The environmental process considers each apex or non-terminal node of the developing tier as the center of a circular leaf cluster, and maintains a list of all clusters present. New clusters are added in response to messages received from the plant. All clusters have the same radius ρ , specified in the environmental data file (*cf.* Figure 2). In order to determine the fate of the apices, the environment compares apex positions with leaf cluster positions, and authorizes an apex to grow if it does not fall into an existing leaf cluster, or if it falls into a cluster surrounding an apex with a smaller vigor value.

The plant model. The plant model is expressed as an open L-system. The values of constants are taken from [33].

```
#define r1 0.94 /* contraction ratio and vigor 1 */
#define r2 0.87 /* contraction ratio and vigor 2 */
#define alpha1 24.4 /* branching angle 1 */
#define alpha2 36.9 /* branching angle 2 */
#define phi 138.5 /* divergence angle */
omega: -(90)[F(1)?E(1)A(1)]+(phi)[F(1)?E(1)A(1)]
      +(phi)[F(1)?E(1)A(1)]+(phi)[F(1)?E(1)A(1)]
      +(phi)[F(1)?E(1)A(1)]

p1: ?E(x) < A(v) : x == 1 ->
      [+ (alpha2)F(v*r2)?E(r2)A(v*r2)] - (alpha1)F(v*r1)?E(r1)A(v*r1)
p2: ?E(x) -> epsilon
```

The axiom ω specifies the initial structure as a whorl of five branch segments F . The divergence angle φ between consecutive segments is equal to 138.5° . Each segment is terminated by a communication symbol $?E$ followed by an apex A . In addition, two branches include module $/$, which changes the directions at which subsequent branches will be issued (left *vs.* right) by rotating the apex 180° around the segment axis.

Production p_1 describes the operation of the apices. If the value of parameter x returned by a communication module $?E$ is not 1, the associated apex will remain inactive (do nothing). Otherwise the apex will produce a pair of new branch segments at angles α_1 and α_2 with respect to the mother segment. Constants r_1 and r_2 determine the lengths of the daughter segments as fractions of the length of their mother segment. The values r_1 and r_2 are also passed to the process simulating the environment using communication modules $?E$. Communication modules created in the previous derivation step are no longer needed and are removed by production p_2 .

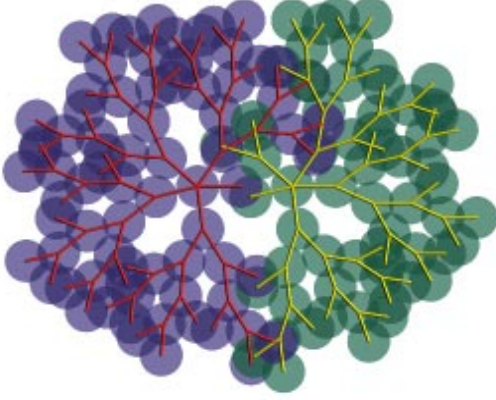


Figure 5: Competition for space between two tiers of branches simulated using the Honda model

Simulation. Figure 5 illustrates the competition for space between two tiers developing next to each other. The extent of branching in each tier is limited by collisions between its apices and its own or the neighbor's leaf clusters. The limited growth of each structure in the direction of its neighbor illustrates the phenomenon of *morphological plasticity*, or adaptation of the form of plants to their environment [17].

6 A MODEL OF FORAGING IN CLONAL PLANTS

Background. Foraging (propagation) patterns in clonal plants provide another excellent example of response to crowding. A clonal plant spreads by means of horizontal stem segments (*spacers*), which form a branching structure that grows along the ground and connects individual plants (*ramets*) [3]. Each ramet consists of a leaf supported by an upright stem and one or more buds, which may give rise to further spacers and ramets. Their gradual death, after a certain amount of time, causes gradual separation of the whole structure (the *clone*) into independent parts.

Following the surface of the soil, clonal plants can be captured using models operating in two dimensions [5], and in that respect resemble *Terminalia* tiers. We propose a model of a hypothetical plant that responds to favorable environmental conditions (high local intensity of light) by more extensive branching and reduced size of leaves (allowing for more dense packing of ramets). It has been inspired by a computer model of clover outlined by Bell [4], the analysis of responses of clonal plants to the environment presented by Dong [17], and the computer models and descriptions of vegetative multiplication of plants involving the death of intervening connections by Room [47].

Communication specification. The plant sends messages to the environment that include turtle position and two parameters associated with the communications symbol, $?E(\text{type}, x)$. The first parameter is equal to 0, 1, or 2, and determines the type of exchanged information as follows:

- The message $?E(0, x)$ represents a request for the light intensity (irradiance [14]) at the position of the communication module.

The environment responds by setting x to the intensity of incoming light, ranging from 0 (no light) to 1 (full light).

- The message $?E(1, x)$ notifies the environment about the creation of a ramet with a leaf of radius x at the position of the communication module. No output is generated by the environment in response to this message.
- The message $?E(2, x)$ notifies the environment about the death of a ramet with a leaf of radius x at the position of the communication module. Again, no output is generated by the environment.

The model of the environment. The purpose of the environment process is to determine light intensity at the locations requested by the plant. The ground is divided into patches (specified as a raster image using a paint program), with different light intensities assigned to each patch. In the absence of shading, these intensities are returned by the environmental process in response to messages of type 0. To consider shading, the environment keeps track of the set of ramets, adding new ramets in response to a messages of type 1, and deleting dead ramets in response to messages of type 2. If a sampling point falls in an area occupied by a ramet, the returned light intensity is equal to 0 (leaves are assumed to be opaque, and located above the sampling points).

The plant model. The essential features of the plant model are specified by the following open L-system.

```
#define  $\alpha$  45          /* branching angle */
#define MinLight 0.1    /* light intensity threshold */
#define MaxAge 20      /* lifetime of ramets and spacers */
#define Len 2.0        /* length of spacers */
#define ProbB(x)      (0.12+x*0.42)
#define ProbR(x)      (0.03+x*0.54)
#define Radius(x)      (sqrt(15-x*5)/ $\pi$ )
 $\omega$ : A(1)?E(0,0)

p1: A(dir) > ?E(0,x) : x >= MinLight
      → R(x)B(x,dir)F(Len,0)A(-dir)?E(0,0)
p2: A(dir) > ?E(0,x) : x < MinLight →  $\varepsilon$ 

p3: B(x,dir) → [( $\alpha$ *dir)F(Len,0)A(-dir)?E(0,0)] : ProbB(x)
p4: B(x,dir) →  $\varepsilon$  : 1-ProbB(x)

p5: R(x) → [ @o(Radius(x),0)?E(1,Radius(x))] : ProbR(x)
p6: R(x) →  $\varepsilon$  : 1-ProbR(x)

p7: @o(radius,age): age < MaxAge → @o(radius,age+1)
p8: @o(radius,age): age == MaxAge → ?E(2,radius)

p9: F(len,age): age < MaxAge → F(len,age+1)
p10: F(len,age): age == MaxAge → f(len)

p11: ?E(type,x) →  $\varepsilon$ 
```

The initial structure specified by the axiom ω consists of an apex A followed by the communication module $?E$. If the intensity of light x reaching an apex is insufficient (below the threshold *MinLight*), the apex dies (production p_2). Otherwise, the apex creates a ramet initial R (i.e., a module that will yield a ramet), a branch initial B , a spacer F , and a new apex A terminated by communication module $?E$ (production p_1). The parameter *dir*, valued either 1 or -1, controls the direction of branching. Parameters of the spacer module specify its length and age.

A branch initial B may create a lateral branch with its own apex A and communication module $?E$ (production p_3), or it may die and disappear from the system (production p_4). The probability of survival is an increasing linear function $Prob_B$ of the light intensity x that has reached the mother apex A in the previous derivation step. A similar stochastic mechanism describes the production of a ramet by the ramet initial R (productions p_5 and p_6), with the probability of ramet formation controlled by an increasing linear function $Prob_R$. The ramet is represented as a circle $@o$; its radius is a decreasing function $Radius$ of the light intensity x . As in the case of spacers, the second parameter of a ramet indicates its age, initially set to 0. The environment is notified about the creation of the ramet using a communication module $?E$.

The subsequent productions describe the aging of spacers (p_7) and ramets (p_9). Upon reaching the maximum age $MaxAge$, a ramet is removed from the system and a message notifying the environment about this fact is sent by the plant (p_8). The death of the spacers is simulated by replacing spacer modules F with invisible line segments f of the same length. This replacement maintains the relative position of the remaining elements of the structure. Finally, production p_{11} removes communication modules after they have performed their tasks.

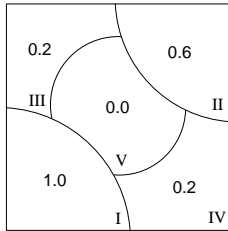


Figure 6: Division of the ground into patches

Simulations. Division of the ground into patches used in the simulations is shown in Figure 6. Arabic numerals indicate the intensity of incoming light, and Roman numerals identify each patch. The development of a clonal plant assuming this division is illustrated in Figure 7. As an extension of the basic model discussed above, the length of the spacers and the magnitude of the branching angle have been varied

using random functions with a normal distribution. Ramets have been represented as trifoliate leaves.

The development begins with a single ramet located in relatively good (light intensity 0.6) patch II at the top right corner of the growth area (Figure 7, step 9 of the simulation). The plant propagates through the unfavorable patch III without producing many branches and leaves (step 26), and reaches the best patch I at the bottom left corner (step 39). After quickly spreading over this patch (step 51), the plant searches for further favorable areas (step 62). The first attempt to reach patch II fails (step 82). The plant tries again, and this time succeeds (steps 101 and 116). Light conditions in patch II are not sufficient, however, to sustain the continuous presence of the plant (step 134). The colony disappears (step 153) until the patch is reached again by a new wave of propagation (steps 161 and 182).

The sustained occupation of patch I and the repetitive invasion of patch II represent an emerging behavior of the model, difficult to predict without running simulations. Variants of this model, including other branching architectures, responses to the environment, and layouts of patches in the environment, would make it possible to analyze different foraging strategies of clonal plants. A further extension could replace the empirical assumptions regarding plant responses with a more detailed simulation of plant physiology (for example, including production of photosynthates and their transport and partition between ramets). Such physiological models could provide insight into the extent to which the foraging patterns optimize plants' access to resources [17].

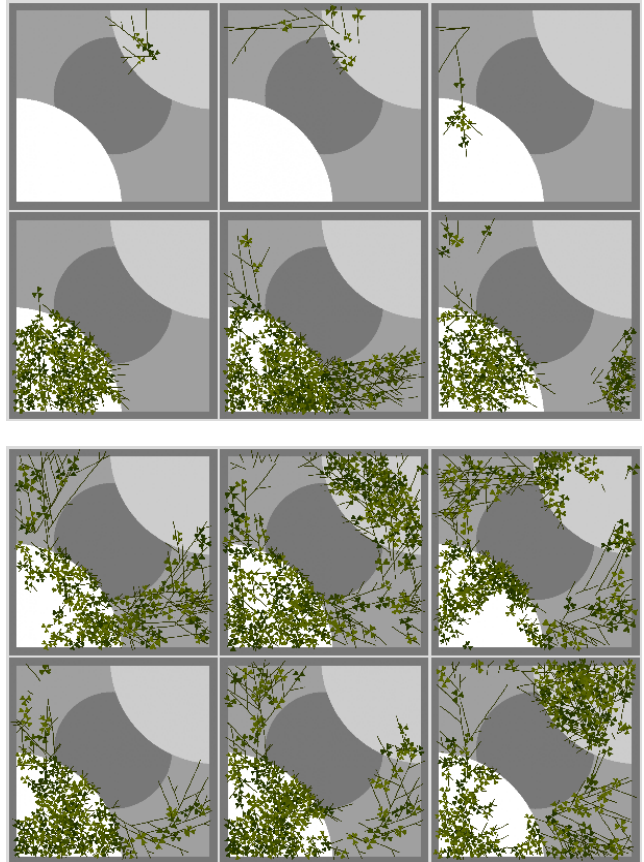


Figure 7: Development of a hypothetical clonal plant simulated using an extension of L-system 3. The individual images represent structures generated in 9, 26, 39, 51, 62, and 82 derivation steps (top), followed by structures generated in 101, 116, 134, 153, 161, and 182 steps (bottom).

7 A MODEL OF ROOT DEVELOPMENT

Background. The development of roots provides many examples of complex interactions with the environment, which involve mechanical properties, chemical reactions, and transport mechanisms in the soil. In particular, the main root and the rootlets absorb water from the soil, locally changing its concentration (volume of water per unit volume of soil) and causing water motion from water-rich to depleted regions [24]. The tips of the roots, in turn, follow the gradient of water concentration [12], thus adapting to the environment modified by their own activities.

Below we present a simplified implementation of the model of root development originally proposed by Clausnitzer and Hopmans [12]. We assume a more rudimentary mechanism of water transport, namely diffusion in a uniform medium, as suggested by Liddell and Hansen [37]. The underlying model of root architecture is similar to that proposed by Diggle [16]. For simplicity, we focus on model operation in two-dimensions.

Communication specification. The plant interacts with the environment using communication modules $?E(c, \theta)$ located at the apices of the root system. A message sent to the environment includes the turtle position \vec{P} , the heading vector \vec{H} , the value of

parameter c representing the requested (optimal) water uptake, and the value of parameter θ representing the tendency of the apex to follow the gradient of water concentration. A message returned to the plant specifies the amount of water actually received by the apex as the value of parameter c , and the angle biasing direction of further growth as the value of θ .

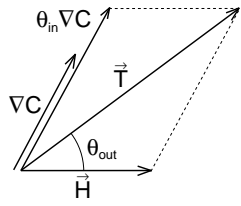


Figure 8: Definition of the biasing angle θ_{out}

The model of the environment.

The environment maintains a field C of water concentrations, represented as an array of the amounts of water in square sampling areas. Water is transported by diffusion, simulated numerically using finite differencing [41]. The environment responds to a request for water from an apex located in an area (i, j) by granting the lesser of the

values requested and available at that location. The amount of water in the sampled area is then decreased by the amount received by the apex. The environment also calculates a linear combination \vec{T} of the turtle heading vector \vec{H} and the gradient of water concentration ∇C (estimated numerically from the water concentrations in the sampled area and its neighbors), and returns an angle θ between the vectors \vec{T} and \vec{H} (Figure 8). This angle is used by the plant model to bias turtle heading in the direction of high water concentration.

The root model. The open L-system representing the root model makes use of arrays that specify parameters for each branching order (main axis, its daughter axes, *etc.*). The parameter values are loosely based on those reported by Clausnitzer and Hopmans [12].

```
#define N 3 /* max. branching order + 1 */
Define: { array
Req[N] = {0.1, 0.4, 0.05}, /* requested nutrient intake */
MinReq[N] = {0.01, 0.06, 0.01}, /* minimum nutrient intake */
ElRate[N] = {0.55, 0.25, 0.55}, /* maximum elongation rate */
MaxLen[N] = {200, 5, 0.8}, /* maximum branch length */
Sens[N] = {10, 0, 0}, /* sensitivity to gradient */
Dev[N] = {30, 75, 75}, /* deviation in heading */
Del[N-1] = {30, 60}, /* delay in branch growth */
BrAngle[N-1] = {90, 90}, /* branching angle */
BrSpace[N-1] = {1, 0.5} /* distance between branches */
}
```

```
 $\omega$ : A(0,0,0)?E(Req[0],Sens[0])
```

```
 $p_1$ : A(n,s,b) > ?E(c, $\theta$ ) : (s > MaxLen[n]) || (c < MinReq[n])  $\rightarrow \epsilon$ 
```

```
 $p_2$ : A(n,s,b) > ?E(c, $\theta$ ) :
(n >= N-1) || (b < BrSpace[n]) {h=c/Req[n]*ElRate[n];
 $\rightarrow$  +(nran( $\theta$ ,Dev[n]))F(h) A(n,s+h,b+h)?E(Req[n],Sens[n])
```

```
 $p_3$ : A(n,s,b) > ?E(c, $\theta$ ) :
(n < N-1) && (b >= BrSpace[n]) {h=c/Req[n]*ElRate[n];
 $\rightarrow$  +(nran( $\theta$ ,Dev[n]))B(n,0)F(h)
/(180)A(n,s+h,h)?E(Req[n],Sens[n])
```

```
 $p_4$ : B(n,t) : t < Del[n]  $\rightarrow$  B(n,t+1)
```

```
 $p_5$ : B(n,t) : t >= Del[n]
 $\rightarrow$  [(BrAngle[n])A(n+1,0,0)?E(Req[n+1],Sens[n+1])]
```

```
 $p_6$ : ?E(c, $\theta$ )  $\rightarrow \epsilon$ 
```

The development starts with an apex A followed by a communication module $?E$. The parameters of the apex represent the branch order (0 for the main axis, 1 for its daughter axes, *etc.*), current axis length, and distance (along the axis) to the nearest branching point.



Figure 9: A two-dimensional model of a root interacting with water in soil. Background colors represent concentrations of water diffusing in soil (blue: high, black: low). The initial and boundary values have been set using a paint program.

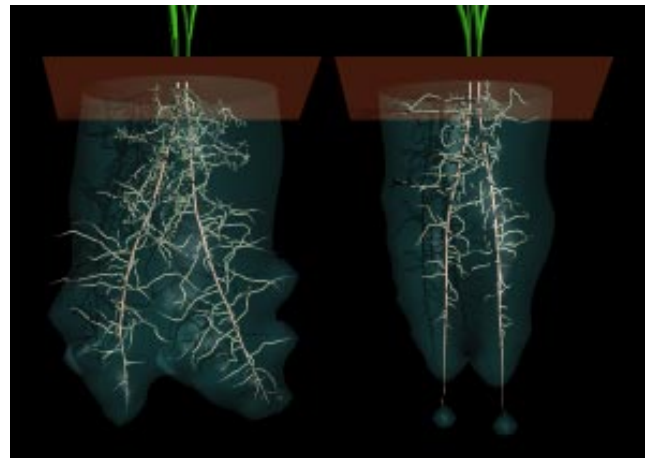


Figure 10: A three-dimensional extension of the root model. Water concentration is visualized by semi-transparent iso-surfaces [55] surrounding the roots. As a result of competition for water, the roots grow away from each other. The divergence between their main axes depends on the spread of the rootlets, which grow faster on the left than on the right.

Productions p_1 to p_3 describe possible fates of the apex as described below.

If the length s of a branch axis exceeds a predefined maximum value $MaxLen[n]$ characteristic to the branch order n , or the amount of water c received by the apex is below the required minimum $MinReq[n]$, the apex dies, terminating the growth of the axis (production p_1).

If the branch order n is equal to the maximum value assumed in the model ($N - 1$), or the distance b to the closest branching point on the axis is less than the threshold value $BrSpace[n]$, the apex adds a new segment F to the axis (production p_2). The length h of F is the product of the nominal growth increment $ElRate[n]$ and the ratio of the amount of water received by the apex c to the amount requested $Req[n]$. The new segment is rotated with respect to its predecessor by an angle $nran(\theta, Dev[n])$, where $nran$ is a random function with a normal distribution. The mean value θ , returned by the environment, biases the direction of growth

towards regions of higher water concentration. The standard deviation $Dev[n]$ characterizes the tendency of the root apex to change direction due to various factors not included explicitly in the model.

If the branch order n is less than the maximum value assumed in the model ($N - 1$), and the distance b to the closest branching point on the axis is equal to or exceeds the threshold value $BrSpace[n]$, the apex creates a new branch initial B (production p_3). Other aspects of apex behavior are the same as those described by production p_2 .

After the delay of $Del[n]$ steps (production p_4), the branch initial B is transformed into an apex A followed by the communication module $?E$ (production p_5), giving rise to a new root branch. Production p_6 removes communication modules that are no longer needed.

Simulations. A sample two-dimensional structure obtained using the described model is shown in Figure 9. The apex of the main axis follows the gradient of water concentration, with small deviations due to random factors. The apices of higher-order axes are not sensitive to the gradient and change direction at random, with a larger standard deviation. The absorption of water by the root and the rootlets decreases water concentration in their neighborhood; an effect that is not fully compensated by water diffusion from the water-rich areas. Low water concentration stops the development of some rootlets before they have reached their potential full length.

Figure 10 presents a three-dimensional extension of the previous model. As a result of competition for water, the main axes of the roots diverge from each other (left). If their rootlets grow more slowly, the area of influence of each root system is smaller and the main axes are closer to each other (right). This behavior is an emergent property of interactions between the root modules, mediated by the environment.

8 MODELS OF TREES CONTROLLED BY LIGHT

Background. Light is one of the most important factors affecting the development of plants. In the essentially two-dimensional structures discussed in Section 5, competition for light could be considered in a manner similar to collision detection between leaves and apices. In contrast, competition for light in three-dimensional structures must be viewed as long-range interaction. Specifically, shadows cast by one branch may affect other branches at significant distances.

The first simulations of plant development that take the local light environment into account are due to Greene [25]. He considered the entire sky hemisphere as a source of illumination and computed the amount of light reaching specific points of the structure by casting rays towards a number of points on the hemisphere. Another approach was implemented by Kanamaru *et al.* [35], who computed the amount of light reaching a given sampling point by considering it a center of projection, from which all leaf clusters in a tree were projected on a surrounding hemisphere. The degree to which the hemisphere was covered by the projected clusters indicated the amount of light received by the sampling point. In both cases, the models of plants responded to the amount and the direction of light by simulating heliotropism, which biased the direction of growth towards the vector of the highest intensity of incoming light. Subsequently, Chiba *et al.* extended the models by Kanamaru *et al.* using more involved tree models that included a mechanism simulating the flow of hypothetical endogenous information within the tree [10, 11]. A biologically better justified model, formulated

in terms of production and use of photosynthates by a tree, was proposed by Takenaka [52]. The amount of light reaching leaf clusters was calculated by sampling a sky hemisphere, as in the work by Greene. Below we reproduce the main features of the Takenaka's model using the formalism of open L-systems. Depending on the underlying tree architecture, it can be applied to synthesize images of deciduous and coniferous trees. We focus on a deciduous tree, which requires a slightly smaller number of productions.

Communication specification. The plant interacts with the environment using communication modules $?E(r)$. A message sent by the plant includes turtle position \vec{P} , which represents the center of a spherical leaf cluster, and the value of parameter r , which represents the cluster's radius. The environment responds by setting r to the flux [14] of light from the sky hemisphere, reaching the cluster.

The model of the environment. Once all messages describing the current distribution of leaves on a tree have been received, the environmental process computes the extent of the tree in the x , y , and z directions, encompasses the tree in a tight grid ($32 \times 32 \times 32$ voxels in our simulations), and allocates leaf clusters to voxels to speed up further computations. The environmental process then estimates the light flux Φ from the sky hemisphere reaching each cluster (shadows cast by the branches are ignored). To this end, the hemisphere is represented by a set of directional light sources S (9 in the simulations). The flux densities (radiosities) B of the sources approximate the non-uniform distribution of light from the sky (*cf.* [52]). For each leaf cluster L_i and each light source S , the environment determines the set of leaf clusters L_j that may shade L_i . This is achieved by casting a ray from the center of L_i in the direction of S and testing for intersections with other clusters (the grid accelerates this process). In order not to miss any clusters that may partially occlude L_i , the radius of each cluster L_j is increased by the maximum value of cluster radius r_{max} .

To calculate the flux reaching cluster L_i , this cluster and all clusters L_j that may shade it according to the described tests are projected on a plane P perpendicular to the direction of light from the source S . The impact of a cluster L_j on the flux Φ reaching cluster L_i is then computed according to the formula:

$$\Phi = (A_i - A_{ij}) B + A_{ij} \tau B$$

where A_i is the area of the projection of L_i on P , A_{ij} is the area of the intersection between projections of L_i and L_j , and τ is the light transmittance through leaf cluster L_j (equal to 0.25 in the simulations). If several clusters L_j shade L_i , their influences are multiplied. The total flux reaching cluster L_i is calculated as the sum of the fluxes received from each light source S .

The plant model. In addition to the communication module $?E$, the plant model includes the following types of modules:

- Apex $A(vig, del)$. Parameter vig represents vigor, which determines the length of branch segments (internodes) and the diameter of leaf clusters produced by the apex. Parameter del is used to introduce a delay, needed for propagating products of photosynthesis through the tree structure between consecutive stages of development (years).
- Leaf $L(vig, p, age, del)$. Parameters denote the leaf radius vig , the amount of photosynthates produced in unit time according to the leaf's exposure to light p , the number of years for which a leaf has appeared at a given location age , and the delay del , which plays the same role as in the apices.
- Internode $F(vig)$. Consistent with the turtle interpretation, the

parameter vig indicates the internode length.

- Branch width symbol $!(w, p, n)$, also used to carry the endogenous information flow. The parameters determine: the width of the following internode w , the amount of photosynthates reaching the symbol's location p , and the number of terminal branch segments above this location n .

The corresponding L-system is given below.

```

#define phi 137.5 /* divergence angle */
#define alpha0 5 /* direction change - no branching */
#define alpha1 20 /* branching angle - main axis */
#define alpha2 32 /* branching angle - lateral axis */
#define W 0.02 /* initial branch width */
#define VD 0.95 /* apex vigor decrement */
#define Del 30 /* delay */
#define LS 5 /* how long a leaf stays */
#define LP 8 /* full photosynthate production */
#define LM 2 /* leaf maintenance */
#define PB 0.8 /* photosynthates needed for branching */
#define PG 0.4 /* photosynthates needed for growth */
#define BM 0.32 /* branch maintenance coefficient */
#define BE 1.5 /* branch maintenance exponent */
#define Nmin 25 /* threshold for shedding */
Consider: ?E[]!L /* for context matching */
omega: !(W,1,1)F(2)L(1,LP,0,0)A(1,0)[!(0,0,0)]!(W,0,1)

p1: A(vig,del) : del<Del -> A(vig,del+1)
p2: L(vig,p,age,del) : (age<LS)&&(del<Del-1) -> L(vig,p,age,del+1)
p3: L(vig,p,age,del) : (age<LS)&&(del==Del-1)
    -> L(vig,p,age,del+1)?E(vig*0.5)
p4: L(vig,p,age,del) > ?E(r) : (age<LS) && (r*LP>=LM)
    && (del == Del) -> L(vig,LP*r-LM,age+1,0)
p5: L(vig,p,age,del) > ?E(r) : ((age == LS)|(r*LP<=LM))
    && (del == Del) -> L(0,0,LS,0)
p6: ?E(r) < A(vig,del) : *LP-LM>PB {vig=vig*VD;}
    -> /((phi)+(alpha2))!(W,-PB,1)F(vig)L(vig,LP,0,0)A(vig,0)
    [!(0,0,0)]!(W,0,1)
    -(alpha1)!(W,0,1)F(vig)L(vig,LP,0,0)/A(vig,0)
p7: ?E(r) < A(vig,del) : r*LP-LM > PG {vig=vig*VD;}
    -> /((phi)-(alpha0))[(0,0,0)]
    !(W,-PG,1)F(vig)L(vig,LP,0,0)A(vig,0)
p8: ?E(r) < A(vig,del) : r*LP-LM <= PG -> A(vig,0)
p9: ?E(r) -> epsilon
p10: !(w0,p0,n0) > L(vig,pL,age,del) [!(w1,p1,n1)]!(w2,p2,n2) :
    {w=(w1^2+w2^2)^0.5; p=p1+p2+pL-BM*(w/W)^BE;}
    (p>0) || (n1+n2 >=Nmin) -> !(w,p,n1+n2)
p11: !(w0,p0,n0) > L(vig,pL,age,del) [!(w1,p1,n1)]!(w2,p2,n2)
    -> !(w0,0,0)L%

```

The simulation starts with a structure consisting of a branch segment F , supporting a leaf L and an apex A (axiom ω). The first branch width symbol $!$ defines the segment width. Two additional symbols $!$ following the apex create “virtual branches,” needed to provide proper context for productions p_{10} and p_{11} . The tree grows in stages, with the delay of $Del + 1$ derivation steps between consecutive stages introduced by production p_1 for the apices and p_2 for the leaves. Immediately before each new growth stage, communication symbols are introduced to inform the environment about the location and size of the leaf clusters (p_3). If the flux r returned by the environment results in the production of photosynthates $r * LP$

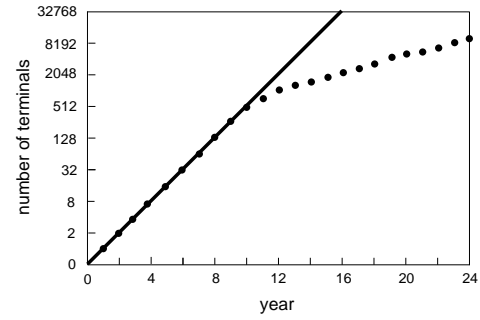


Figure 11: The number of terminal branch segments resulting from unrestricted bifurcation of apices (continuous line), compared to the number of segments generated in a simulation (isolated points)

exceeding the amount LM needed to maintain a cluster, it remains in the structure (p_4). Otherwise it becomes a liability to the tree and dies (p_5). Another condition to production p_5 prevents a leaf from occupying the same location for more than LS years.

The flux r also determines the fate of the apex, captured by productions p_6 to p_8 . If the amount of photosynthates $r * LP - LM$ transported from the nearby leaf exceeds a threshold value PB , the apex produces two new branches (p_6). The second parameter in the first branch symbol $!$ is set to $-PB$, to subtract the amount of photosynthates used for branching from the amount that will be transported further down. The length of branch segments vig is reduced with respect to the mother segment by a predefined factor VD , reflecting a gradual decrease in the vigor of apices with age. The branch width modules $!$ following the first apex A are introduced to provide context required by productions p_{10} and p_{11} , as in the axiom.

If the amount of photosynthates $r * LP - LM$ transported from the leaf is insufficient to produce new branches, but above the threshold PG , the apex adds a new segment F to the current branch axis without creating a lateral branch (p_7). Again, a virtual branch containing the branch width symbol $!$ is being added to provide context for productions p_{10} and p_{11} .

If the amount of photosynthates is below PG , the apex remains dormant (p_8). Communication modules no longer needed are removed from the structure (p_9).

Production p_{10} captures the endogenous information flow from leaves and terminal branch segments to the base of the tree. First, it determines the radius w of the mother branch segment as a function of the radii w_1 and w_2 of the supported branches:

$$w = \sqrt{w_1^2 + w_2^2}.$$

Thus, a cross section of the mother segment has an area equal to the sum of cross sections of the supported segments, as postulated in the literature [40, 46]. Next, production p_{10} calculates the flow p of photosynthates into the mother segment. It is defined as the sum of the flows p_L , p_1 and p_2 received from the associated leaf L and from both daughter branches, decreased by the amount $BM * (w/W)^{BE}$ representing the cost of maintaining the mother segment. Finally, production p_9 calculates the number of terminal branch segments n supported by the mother segment as the sum of the numbers of terminal segments supported by the daughter branches, n_1 and n_2 .

Production p_{10} takes effect if the flow p is positive (the branch is not a liability to the tree), or if the number n of supported terminals



Figure 12: A tree model with branches competing for access to light, shown without the leaves



Figure 13: A climbing plant growing on the tree from the previous figure

is above a threshold N_{min} . If these conditions are not satisfied, production p_{11} removes (sheds) the branch from the tree using the cut symbol %.

Simulations. The competition for light between tree branches is manifested by two phenomena: reduced branching or dormancy



Figure 14: A model of deciduous trees competing for light. The trees are shown in the position of growth (top) and moved apart (bottom) to reveal the adaptation of crown geometry to the presence of the neighbor tree.

of apices in unfavorable local light conditions, and shedding of branches which do not receive enough light to contribute to the whole tree. Both phenomena limit the extent of branching, thus controlling the density of the crown. This property of the model is supported by the simulation results shown in Figure 11. If the growth was unlimited (production p_6 was always chosen over p_7 and p_8), the number of terminal branch segments would double every year. Due to the competition for light, however, the number of terminal segments observed in an actual simulation increases more slowly. For related statistics using a different tree architecture see [52].

A tree image synthesized using an extension of the presented model is shown in Figure 12. The key additional feature is a gradual reduction of the branching angle of a young branch whose sister branch has been shed. As the result, the remaining branch assumes the role of the leading shoot, following the general growth direction of its supporting segment. Branch segments are represented as texture-mapped generalized cylinders, smoothly connected at the branching points (cf. [6]). The bark texture was created using a paint program.

As an illustration of the flexibility of the modeling framework presented in this paper, Figure 13 shows the effect of seeding a hypothetical climbing plant near the same tree. The plant follows the surface of the tree trunk and branches, and avoids excessively dense



Figure 15: A model of coniferous trees competing for light. The trees are shown in the position of growth (top) and moved apart (bottom).

colonization of any particular area. Thus, the model integrates several environmentally-controlled phenomena: the competition of tree branches for light, the following of surfaces by a climbing plant, and the prevention of crowding as discussed in Section 6. Leaves were modeled using cubic patches (*cf.* [46]).

In the simulations shown in Figure 14 two trees described by the same set of rules (younger specimens of the tree from Figure 12) compete for light from the sky hemisphere. Moving the trees apart after they have grown reveals the adaptation of their crowns to the presence of the neighbor tree. This simulation illustrates both the necessity and the possibility of incorporating the adaptive behavior into tree models used for landscape design purposes.

The same phenomenon applies to coniferous trees, as illustrated in Figure 15. The tree model is similar to the original model by Takenaka [52] and can be viewed as consisting of approximately horizontal tiers (as discussed in Section 5) produced in sequence by the apex of the tree stem. The lower tiers are created first and therefore potentially can spread more widely than the younger tiers higher up (the *phase effect* [46]). This pattern of development is affected by the presence of the neighboring tree: the competition for light prevents the crowns from growing into each other.

The trees in Figure 15 retain branches that do not receive enough light. In contrast, the trees in the stand presented in Figure 16 shed

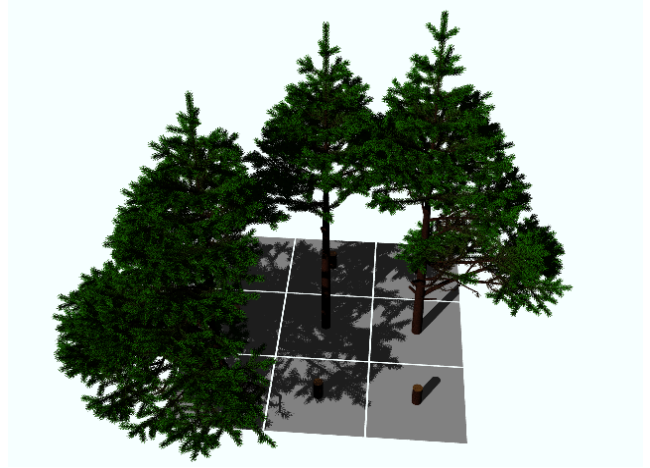


Figure 16: Relationship between tree form and its position in a stand.

branches that do not contribute photosynthates to the entire tree, using the same mechanism as described for the deciduous trees. The resulting simulation reveals essential differences between the shape of the tree crown in the middle of a stand, at the edge, or at the corner. In particular, the tree in the middle retains only the upper part of its crown. In lumber industry, the loss of lower branches is usually a desirable phenomenon, as it reduces knots in the wood and the amount of cleaning that trees require before transport. Simulations may assist in choosing an optimal distance for planting trees, where self-pruning is maximized, yet there is sufficient space between trees too allow for unimpeded growth of trunks in height and diameter.

9 CONCLUSIONS

In this paper, we introduced a framework for the modeling and visualization of plants interacting with their environment. The essential elements of this framework are:

- a system design, in which the plant and the environment are treated as two separate processes, communicating using a standard interface, and
- the language of open L-systems, used to specify plant models that can exchange information with the environment.

We demonstrated the operation of this framework by implementing models that capture collisions between branches, the propagation of clonal plants, the development of roots in soil, and the development of tree crowns competing for light. We found that the proposed framework makes it possible to easily create and modify models spanning a wide range of plant structures and environmental processes. Simulations of the presented phenomena were fast enough to allow interactive experimentation with the models (Table 1).

There are many research topics that may be addressed using the simulation and visualization capabilities of the proposed framework. They include, for instance:

- Fundamental analysis of the role of different forms of information flow in plant morphogenesis (in particular, the relationship

Fig.	Number of		Derivation		Time ^a	
	branch segments	leaf clusters	steps	yrs	sim.	render.
5	138	140	5	5	1 s	1 s
7	786	229	182	NA	50 s	2 s
9	4194	34 ^b	186	NA	67 s	3 s
10	37228	448 ^b	301	NA	15 min	70 s
12	22462	19195	744	24	22 min	13 s ^c
15	13502	3448	194	15	4 min	8 s ^d

^aSimulation and rendering using OpenGL on a 200MHz/64MB Indigo² Extreme

^bactive apices

^cwithout generalized cylinders and texture mapping

^dbranching structure without needles

Table 1: Numbers of primitives and simulation/rendering times for generating and visualizing selected models

between endogenous and exogenous flow). This is a continuation of the research pioneered by Bell [4] and Honda *et al.* [7, 33].

- Development of a comprehensive plant model describing the cycling of nutrients from the soil through the roots and branches to the leaves, then back to the soil in the form of substances released by fallen leaves.
- Development of models of specific plants for research, crop and forest management, and for landscape design purposes. The models may include environmental phenomena not discussed in this paper, such as the global distribution of radiative energy in the tree crowns, which affects the amount of light reaching the leaves and the local temperature of plant organs.

The presented framework itself is also open to further research. To begin, the precise functional specification of the environment, implied by the design of the modeling framework, is suitable for a formal analysis of algorithms that capture various environmental processes. This analysis may highlight tradeoffs between time, memory, and communication complexity, and lead to programs matching the needs of the model to available system resources in an optimal manner.

A deeper understanding of the spectrum of processes taking place in the environment may lead to the design of a mini-language for environment specification. Analogous to the language of L-systems for plant specification, this mini-language would simplify the modeling of various environments, relieving the modeler from the burden of low-level programming in a general-purpose language. Fleischer and Barr's work on the specification of environments supporting collisions and reaction-diffusion processes [20] is an inspiring step in this direction.

Complexity issues are not limited to the environment, but also arise in plant models. They become particularly relevant as the scope of modeling increases from individual plants to groups of plants and, eventually, entire plant communities. This raises the problem of selecting the proper level of abstraction for designing plant models, including careful selection of physiological processes incorporated into the model and the spatial resolution of the resulting structures.

The complexity of the modeling task can be also addressed at the level of system design, by assigning various components of the model (individual plants and aspects of the environment) to different components of a distributed computing system. The communication

structure should then be redesigned to accommodate information transfers between numerous processes within the system.

In summary, we believe that the proposed modeling methodology and its extensions will prove useful in many applications of plant modeling, from research in plant development and ecology to landscape design and realistic image synthesis.

Acknowledgements

We would like to thank Johannes Battjes, Campbell Davidson, Art Diggle, Heinjo During, Michael Guzy, Naoyoshi Kanamaru, Bruno Moulia, Zbigniew Prusinkiewicz, Bill Remphrey, David Reid, and Peter Room for discussions and pointers to the literature relevant to this paper. We would also like to thank Bruno Andrieu, Mark Hammel, Jim Hanan, Lynn Mercer, Chris Prusinkiewicz, Peter Room, and the anonymous referees for helpful comments on the manuscript. Most images were rendered using the ray tracer rayshade by Craig Kolb. This research was sponsored by grants from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

- [1] AGRAWAL, P. The cell programming language. *Artificial Life* 2, 1 (1995), 37–77.
- [2] ARVO, J., AND KIRK, D. Modeling plants with environment-sensitive automata. In *Proceedings of Ausgraph'88* (1988), pp. 27 – 33.
- [3] BELL, A. *Plant form: An illustrated guide to flowering plants*. Oxford University Press, Oxford, 1991.
- [4] BELL, A. D. The simulation of branching patterns in modular organisms. *Philos. Trans. Royal Society London, Ser. B* 313 (1986), 143–169.
- [5] BELL, A. D., ROBERTS, D., AND SMITH, A. Branching patterns: the simulation of plant architecture. *Journal of Theoretical Biology* 81 (1979), 351–375.
- [6] BLOOMENTHAL, J. Modeling the Mighty Maple. Proceedings of SIGGRAPH '85 (San Francisco, California, July 22-26, 1985), in *Computer Graphics*, 19, 3 (July 1985), pages 305–311, ACM SIGGRAPH, New York, 1985.
- [7] BORCHERT, R., AND HONDA, H. Control of development in the bifurcating branch system of *Tabebuia rosea*: A computer simulation. *Botanical Gazette* 145, 2 (1984), 184–195.
- [8] BORCHERT, R., AND SLADE, N. Bifurcation ratios and the adaptive geometry of trees. *Botanical Gazette* 142, 3 (1981), 394–401.
- [9] CHEN, S. G., CEULEMANS, R., AND IMPENS, I. A fractal based *Populus* canopy structure model for the calculation of light interception. *Forest Ecology and Management* (1993).
- [10] CHIBA, N., OHKAWA, S., MURAOKA, K., AND MIURA, M. Visual simulation of botanical trees based on virtual heliotropism and dormancy break. *The Journal of Visualization and Computer Animation* 5 (1994), 3–15.
- [11] CHIBA, N., OHSHIDA, K., MURAOKA, K., MIURA, M., AND SAITO, N. A growth model having the abilities of growth-regulations for simulating visual nature of botanical trees. *Computers and Graphics* 18, 4 (1994), 469–479.
- [12] CLAUSNITZER, V., AND HOPMANS, J. Simultaneous modeling of transient three-dimensional root growth and soil water flow. *Plant and Soil* 164 (1994), 299–314.
- [13] COHEN, D. Computer simulation of biological pattern generation processes. *Nature* 216 (October 1967), 246–248.

- [14] COHEN, M., AND WALLACE, J. *Radiosity and realistic image synthesis*. Academic Press Professional, Boston, 1993. With a chapter by P. Hanrahan and a foreword by D. Greenberg.
- [15] DE REFFYE, P., HOULLIER, F., BLAISE, F., BARTHELEMY, D., DAUZAT, J., AND AUCLAIR, D. A model simulating above- and below-ground tree architecture with agroforestry applications. *Agroforestry Systems* 30 (1995), 175–197.
- [16] DIGGLE, A. J. ROOTMAP - a model in three-dimensional coordinates of the structure and growth of fibrous root systems. *Plant and Soil* 105 (1988), 169–178.
- [17] DONG, M. *Foraging through morphological response in clonal herbs*. PhD thesis, University of Utrecht, October 1994.
- [18] FISHER, J. B. How predictive are computer simulations of tree architecture. *International Journal of Plant Sciences* 153 (Suppl.) (1992), 137–146.
- [19] FISHER, J. B., AND HONDA, H. Computer simulation of branching pattern and geometry in *Terminalia* (Combretaceae), a tropical tree. *Botanical Gazette* 138, 4 (1977), 377–384.
- [20] FLEISCHER, K. W., AND BARR, A. H. A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In *Artificial Life III*, C. G. Langton, Ed. Addison-Wesley, Redwood City, 1994, pp. 389–416.
- [21] FORD, E. D., AVERY, A., AND FORD, R. Simulation of branch growth in the *Pinaceae*: Interactions of morphology, phenology, foliage productivity, and the requirement for structural support, on the export of carbon. *Journal of Theoretical Biology* 146 (1990), 15–36.
- [22] FORD, H. Investigating the ecological and evolutionary significance of plant growth form using stochastic simulation. *Annals of Botany* 59 (1987), 487–494.
- [23] FRIJTERS, D., AND LINDENMAYER, A. A model for the growth and flowering of *Aster novae-angliae* on the basis of table (1,0)L-systems. In *L-Systems*, G. Rozenberg and A. Salomaa, Eds., Lecture Notes in Computer Science 15. Springer-Verlag, Berlin, 1974, pp. 24–52.
- [24] GARDNER, W. R. Dynamic aspects of water availability to plants. *Soil Science* 89, 2 (1960), 63–73.
- [25] GREENE, N. Voxel space automata: Modeling with stochastic growth processes in voxel space. Proceedings of SIGGRAPH '89 (Boston, Mass., July 31–August 4, 1989), in *Computer Graphics* 23, 4 (August 1989), pages 175–184, ACM SIGGRAPH, New York, 1989.
- [26] GREENE, N. Detailing tree skeletons with voxel automata. SIGGRAPH '91 Course Notes on Photorealistic Volume Modeling and Rendering Techniques, 1991.
- [27] GUZY, M. R. A morphological-mechanistic plant model formalized in an object-oriented parametric L-system. Manuscript, USDA-ARS Salinity Laboratory, Riverside, 1995.
- [28] HANAN, J. Virtual plants — Integrating architectural and physiological plant models. In *Proceedings of ModSim 95* (Perth, 1995), P. Binning, H. Bridgman, and B. Williams, Eds., vol. 1, The Modelling and Simulation Society of Australia, pp. 44–50.
- [29] HANAN, J. S. *Parametric L-systems and their application to the modelling and visualization of plants*. PhD thesis, University of Regina, June 1992.
- [30] HART, J. W. *Plant tropisms and other growth movements*. Unwin Hyman, London, 1990.
- [31] HERMAN, G. T., AND ROZENBERG, G. *Developmental systems and languages*. North-Holland, Amsterdam, 1975.
- [32] HONDA, H. Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology* 31 (1971), 331–338.
- [33] HONDA, H., TOMLINSON, P. B., AND FISHER, J. B. Computer simulation of branch interaction and regulation by unequal flow rates in botanical trees. *American Journal of Botany* 68 (1981), 569–585.
- [34] KAANDORP, J. *Fractal modelling: Growth and form in biology*. Springer-Verlag, Berlin, 1994.
- [35] KANAMARU, N., CHIBA, N., TAKAHASHI, K., AND SAITO, N. CG simulation of natural shapes of botanical trees based on heliotropism. *The Transactions of the Institute of Electronics, Information, and Communication Engineers J75-D-II*, 1 (1992), 76–85. In Japanese.
- [36] KURTH, W. *Growth grammar interpreter GROGRA 2.4: A software tool for the 3-dimensional interpretation of stochastic, sensitive growth grammars in the context of plant modeling. Introduction and reference manual*. Forschungszentrum Waldökosysteme der Universität Göttingen, Göttingen, 1994.
- [37] LIDDELL, C. M., AND HANSEN, D. Visualizing complex biological interactions in the soil ecosystem. *The Journal of Visualization and Computer Animation* 4 (1993), 3–12.
- [38] LINDENMAYER, A. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology* 18 (1968), 280–315.
- [39] LINDENMAYER, A. Developmental systems without cellular interaction, their languages and grammars. *Journal of Theoretical Biology* 30 (1971), 455–484.
- [40] MACDONALD, N. *Trees and networks in biological models*. J. Wiley & Sons, New York, 1983.
- [41] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. *Numerical recipes in C: The art of scientific computing. Second edition*. Cambridge University Press, Cambridge, 1992.
- [42] PRUSINKIEWICZ, P. Visual models of morphogenesis. *Artificial Life* 1, 1/2 (1994), 61–74.
- [43] PRUSINKIEWICZ, P., HAMMEL, M., HANAN, J., AND MĚCH, R. Visual models of plant development. In *Handbook of formal languages*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, Berlin, 1996. To appear.
- [44] PRUSINKIEWICZ, P., AND HANAN, J. L-systems: From formalism to programming languages. In *Lindenmayer systems: Impacts on the theoretical computer science, computer graphics, and developmental biology*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, Berlin, 1992, pp. 193–211.
- [45] PRUSINKIEWICZ, P., JAMES, M., AND MĚCH, R. Synthetic topiary. Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994), pages 351–358, ACM SIGGRAPH, New York, 1994.
- [46] PRUSINKIEWICZ, P., AND LINDENMAYER, A. *The algorithmic beauty of plants*. Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.
- [47] ROOM, P. M. 'Falling apart' as a lifestyle: the rhizome architecture and population growth of *Salvinia molesta*. *Journal of Ecology* 71 (1983), 349–365.
- [48] ROOM, P. M., MAILLETTE, L., AND HANAN, J. Module and metamer dynamics and virtual plants. *Advances in Ecological Research* 25 (1994), 105–157.
- [49] ROZENBERG, G. TOL systems and languages. *Information and Control* 23 (1973), 357–381.
- [50] SACHS, T., AND NOVOPLANSKY, A. Tree from: Architectural models do not suffice. *Israel Journal of Plant Sciences* 43 (1995), 203–212.
- [51] SIPPER, M. Studying artificial life using a simple, general cellular model. *Artificial Life* 2, 1 (1995), 1–35.
- [52] TAKENAKA, A. A simulation model of tree architecture development based on growth response to local light environment. *Journal of Plant Research* 107 (1994), 321–330.
- [53] ULAM, S. On some mathematical properties connected with patterns of growth of figures. In *Proceedings of Symposia on Applied Mathematics* (1962), vol. 14, American Mathematical Society, pp. 215–224.
- [54] WEBER, J., AND PENN, J. Creation and rendering of realistic trees. Proceedings of SIGGRAPH '95 (Los Angeles, California, August 6–11, 1995), pages 119–128, ACM SIGGRAPH, New York, 1995.
- [55] WYVILL, G., MCPHEETERS, C., AND WYVILL, B. Data structure for soft objects. *The Visual Computer* 2, 4 (February 1986), 227–234.