

Art and Science for Life: Designing and Growing Virtual Plants with L-systems

Przemyslaw Prusinkiewicz
Department of Computer Science, University of Calgary
Calgary, Alberta, Canada T2N 1N4
e-mail: pwp@cpsc.ucalgary.ca

Abstract

Virtual plants are computer models that recreate the structure and simulate the development of plants. Virtual plant modeling is an interdisciplinary area combining mathematical formalisms, biological knowledge, and computer graphics techniques. An important modeling method is based on the theory of Lindenmayer systems (L-systems). At present, L-system models make it possible to: (a) accurately reproduce the structure and development of plants; (b) show how architectural parameters (e.g., branching angles, elongation rates, vigor of branches) affect the appearance of plants; (c) simulate plant physiology and investigate the effects of manipulations (e.g., pruning, bonsai techniques) or different external conditions (e.g., local light microclimate, water availability, crowding) on plant development; and (d) simulate plants not only in isolation, but also in their ecological contexts. Currently constructed research models also address the genetic basis of plant form. In horticulture, the models are potentially useful as an exploration tool, indicating desirable directions of breeding and manipulating ornamental plants for maximum visual impact, and fruit plants for maximum yield. Other applications include fundamental research and teaching of biology, and landscape design. This paper describes the current state of the L-system-based modeling methodology as supported by L-studio and Virtual Laboratory, plant modeling software developed at the University of Calgary.

Reference

P. Prusinkiewicz: Art and science for life: Designing and growing virtual plants with L-systems. In C. Davidson and T. Fernandez (Eds:) *Nursery Crops: Development, Evaluation, Production and Use: Proceedings of the XXVI International Horticultural Congress. Acta Horticulturae* 630, pp. 15–28.

Art and Science for Life: Designing and Growing Virtual Plants with L-systems

Przemyslaw Prusinkiewicz
Department of Computer Science
University of Calgary
Calgary, Alberta
Canada

Keywords: modeling and visualization of plants, L-studio, Virtual Laboratory

Abstract

Virtual plants are computer models that recreate the structure and simulate the development of plants. Virtual plant modeling is an interdisciplinary area combining mathematical formalisms, biological knowledge, and computer graphics techniques. An important modeling method is based on the theory of Lindenmayer systems (L-systems). At present, L-system models make it possible to: (a) accurately reproduce the structure and development of plants; (b) show how architectural parameters (e.g., branching angles, elongation rates, vigor of branches) affect the appearance of plants; (c) simulate plant physiology and investigate the effects of manipulations (e.g., pruning, bonsai techniques) or different external conditions (e.g., local light microclimate, water availability, crowding) on plant development; and (d) simulate plants not only in isolation, but also in their ecological contexts. Currently constructed research models also address the genetic basis of plant form. In horticulture, the models are potentially useful as an exploration tool, indicating desirable directions of breeding and manipulating ornamental plants for maximum visual impact, and fruit plants for maximum yield. Other applications include fundamental research and teaching of biology, and landscape design. This paper describes the current state of the L-system-based modeling methodology as supported by L-studio and Virtual Laboratory, plant modeling software developed at the University of Calgary.

INTRODUCTION

The modeling of plants has the captivating appeal of reproducing the visual beauty of nature while providing insights into the way nature works. This interplay between science and art is well rooted in history and can be traced to Leonardo da Vinci (1970), who formulated his observations of plant structure in mathematical terms as a guide for painting.

Today, the modeling of plants is a highly interdisciplinary area. Botany and applied plant sciences are at the root of many approaches to model construction, and an obvious domain of model applications. Mathematics, statistics, and theoretical computer science contribute to the modeling methodology. Substantial progress in the modeling of plants is also due to computer graphics. Realistic presentation adds credibility to biological models by making it easier to visually compare them to nature (Prusinkiewicz, 1998a). Furthermore, visual appearance is of primary importance in such applications as the modeling of ornamental plants and computer-assisted design of landscapes and gardens.

Models may correspond to various levels of plant organization (Thornley and Johnson, 1990). The architectural level (Hallé et al., 1978) is particularly conducive to modeling, because it reflects the modular organization of plants (Harper, 1977). Modules represent repeating components of plant structure, such as flowers, leaves, and internodes (c.f. Bell, 1991). While the total number of modules in a plant may be large, the number of different module types is relatively small. Consequently, a plant can be concisely described by characterizing the behavior of module *types*, rather than individual *occurrences* of each module in the plant. This reduction underlies all existing models of

plant architecture, and is the cornerstone of plant models expressed using L-systems.

L-SYSTEMS

Lindenmayer systems, or L-systems, originated within theoretical biology (Lindenmayer, 1968) and were extensively studied by mathematicians and computer scientists (e.g., Herman and Rozenberg, 1975) before they became an effective modeling tool used for biological simulations and image synthesis purposes (for a historical perspective see Prusinkiewicz, 1999). The formalism of L-systems treats plant development as a sequence of events, in which *predecessor* (parent) modules are replaced by configurations of *successor* (child) modules. These replacements may have a qualitative character, for example when describing the creation of a metamer by an apex, or a quantitative character, for example when describing the elongation of an internode. The rules of replacement are called *productions*. It is assumed that the number of different module types is finite, and all modules of the same type behave in the same manner. Consequently, the number of productions is finite as well, leading to a finite specification of the development of an arbitrarily large structure. In addition to the list of productions, the full L-system description of a plant includes a list of all module types that can be found in a given organism, called the L-system *alphabet*, and a specification of the initial configuration of modules from which the development begins, called the *axiom*. This terminology reflects the relationship between L-systems and theoretical computer science.

The basic concept of L-systems is illustrated in Figure 1, using a stylized model of a compound carrot-like leaf as an example. This model operates on two classes of modules: apices and internodes. According to the first production, an apex yields a branching structure that consists of an apex continuing the main axis, two lateral apices, and two internodes. The second production states that, over the corresponding time interval, an internode doubles its length. The developmental sequence begins with a single apex, and results in an intricate branching structure in spite of the simplicity of this L-system.

L-systems can be expressed symbolically using a mathematical notation or a programming language based on this notation. Several variants of such languages have been developed in the past (see Prusinkiewicz, 1999, for a review, and Prusinkiewicz et al., 2000a; Erstad, 2002; Karwowski and Prusinkiewicz, 2003, for the most recent results). The availability of a programming language is the most distinctive feature of L-system-based modeling systems. The modeling languages make it possible to create models of different types, both descriptive and mechanistic, and simulate the development of a variety of organisms, from multicellular bacteria to herbaceous plants to trees, using the same simulation program with different input files. This approach offers the following benefits:

- The programming effort needed to develop L-system models of specific plants is significantly reduced in comparison to the effort needed to implement the same models “from scratch” in a general-purpose programming language, such as C++ or Java;
- The models can be easily modified during experimentation. These modifications are not limited to the values of numerical parameters, but may also involve fundamental changes in model definition and operation;
- The L-system language makes it possible to document models in a compact and precise manner (for example, in publications).

Information flow between modules plays a crucial role in the control of developmental processes in plants. L-systems provide an elegant mechanism for simulating various forms of this flow. *Lineage* is the transfer of information from a parent module to its children. It is expressed using *context-free* L-system productions, in which the type and parameters of the predecessor modules completely define the successor structures. In contrast, *endogenous* information flow is expressed using *context-sensitive* productions. In this case, the fate of a module being replaced depends not only on the module itself, but also on its neighbors. Context-sensitive productions capture transport

mechanisms in plants, such as the flow of water, hormones, or products of photosynthesis. *Exogenous interaction* results from information transfer through the space in which plants grow. *Environmentally-sensitive* L-systems (Prusinkiewicz et al., 1994) capture the unilateral impact of the environment on plant development. *Open L-systems* (Měch and Prusinkiewicz, 1996) make it possible to express the full bidirectional feedback loop of interactions, in which the environment affects the plant, and the plant reciprocally affects the environment. Examples of such bidirectional interactions include competition for space and light between branches of a tree, and competition for water between roots. All these processes are supported by the modeling software L-studio and the Virtual Laboratory.

L-STUDIO AND THE VIRTUAL LABORATORY

L-studio and the Virtual Laboratory (Vlab) are two related plant modeling packages developed at, and distributed by, the University of Calgary, Canada. They run on Windows and Linux machines, have similar functionality, and support the exchange of models between both systems. The main difference between these systems is the user interface, which is designed to meet the different expectations of Windows or Linux users.

Each of the L-studio and Vlab systems consists of: (a) two L-system-based simulation programs, cpfg and lpfg; (b) a modeling environment that provides auxiliary modeling tools and a graphical interface for creating and manipulating them; (c) a library of programs for simulating environmental processes that affect plant development; (d) a set of sample models; and (e) a graphical browser for organizing and accessing models on both local and remote machines. A sample of a computer screen with L-studio in operation is shown in Figure 2. We outline the main components of both systems below.

Simulation Programs

The simulation programs cpfg and lpfg are at the heart of L-studio and Vlab. Their design has been guided by two key objectives: (a) flexibility, making it possible to model and simulate a wide range of structures and developmental processes in plants, and (b) visual realism of models. To meet the flexibility criterion, the models are specified by the users in L-system-based languages. The cpfg language, developed first, is based on the mathematical notation of L-systems, and makes it possible to specify simple models very concisely. The lpfg language, called L+C, incorporates the L-system notion of productions into the full implementation of the C++ programming language. L+C supports several constructs absent in the cpfg language, such as the possibility of assigning structured parameters to modules, support for user-definable functions, support for productions with alternative successors, and a mechanism for fast transfer of unidirectional information. As a result, L+C is advantageous when developing complex plant models (Figure 3).

For example, the L-systems that generates the compound leaf model of Figure 1 has the following specification in the cpfg language:

```
axiom: A(1)
A(s) --> I(s)[-(45)A(s)][+(45)A(s)]I(s)A(s)
I(s) --> I(2*s)
```

Specification of the same L-system in L+C has the form:

```
module Apex(float);
module Internode(float);
axiom: Apex(1.0);
A(s): {produce Internode(s)
      SB() Left(45) Apex(s) EB()
      SB() Right(45) Apex(s) EB()
      Internode(s) Apex(s) ;}
Internode(s): {produce Internode(2*s) ;}
```

In the case of cpfg, the axes of a branching structure are represented by sequences of letters that indicate the type of each constituent module. Branches are enclosed in square brackets. Auxiliary symbols (letters of the L-system alphabet) and numerically-valued parameters specify attributes of the modeled structure, such as branching angles. The L+C specification is similar in spirit, but the modules can be denoted using full-word identifiers, and must be declared at the beginning of the program. The differences between the cpfg language and L+C become more substantial when constructing complex models.

Based on their input (a number of files including the L-system model and optional data files), cpfg and lpfg create three-dimensional internal representations of a model and project it on the screen. Model visualization is based on the OpenGL graphics library (Woo et al., 1999) and supports standard modeling and rendering techniques developed in computer graphics, such as parametric surfaces, generalized cylinders, and texture mapping (c.f. Foley et. al., 1990). The output may be a static model (which can be interactively rotated and zoomed in by the user) or computer-generated animation that results from visualizing consecutive stages of the simulation (Figure 4). The visualizations may have the form of schematic or realistic representations of branching structures, as well as plots and histograms. The generated images can be output in several raster formats and as PostScript files. The 3D models can also be exported to external rendering and modeling programs. In addition, user-specified quantities can be output to files, allowing for further analysis of the models using external programs. These include general-purpose statistical programs, such as Excel, and AMAPmod, a program dedicated to analyzing plant architectures (Godin et al., 1997; Ferraro et al., 2002).

Both cpfg and lpfg make it possible to construct models with interactive features. The user can interact with the models by selecting a model component with a mouse, and assigning a programmable action to this selection. Sample applications of this feature include: simulation of pruning and grafting (Figure 5), interactive placement of insects or pathogens on selected parts of the plant, selective increase or decrease of the level of detail in the model presentation, and selective display of parameters associated with the model.

The Modeling Environments

In addition to the L-system-based description of the essential aspects of models, cpfg and lpfg require information to control the viewing and animation processes, and to characterize visual attributes of the models, such as the shapes and colors of plant organs. L-studio and the Virtual Laboratory provide user interfaces for specifying this complementary information and transferring it to the simulation programs. The L-studio graphical interface (Prusinkiewicz et al., 2000b) is organized according to the Microsoft MDI (Multiple Document Interface) standard (Petzold, 1998). Different editors can be selected one at a time using tabs associated with the L-studio window (Figure 1). The Virtual Laboratory environment (Mercer et al., 1990; Federl and Prusinkiewicz, 1999) provides a more flexible interface, in which an arbitrary number of editors can be open simultaneously, and additional (e.g., user-developed) programs can easily be incorporated into the modeling framework. In both cases, the available editors include: (a) a *text* editor for creating and manipulating L-system and other textual files that constitute the model; (b) a *control panel manager*, which makes it possible to create user-configurable controls, such as sliders and buttons, for manipulating model parameters; (c) *surface* and *cross-section* editors for defining the shape of individual plant organs; (d) *color* and *material* editors for specifying plant appearance when rendering; and (e) a graphical editor of *functions* of a single variable (Figure 6). Some editors are complemented with *galleries* that make it possible to select a specific object or feature to be edited within a given class (for example, the shape of a petal within the class of organ shapes).

The graphically-specified functions provide a flexible mechanism for defining and manipulating many aspects of the model, for example the growth functions for model components. In practice, these functions are invaluable when augmenting the model with

features that are visually important, yet have been considered of secondary relevance to the modeled process or structure and, therefore, have not been measured. These often include features such as the curvature of plant axes and the orientation of plant organs.

Environmental Programs

An important application area of architectural plant modeling is the study of interactions between plants and their environment. L-studio and vlab make it possible to simulate such interaction using concurrently running, communicating programs that represent the plant and its environment. Several programs for simulating environmentally-mediated phenomena are available in both L-studio and Vlab. They include: programs that model collisions between plant organs and their environment, programs for calculating direct and diffuse light distribution in a plant, and programs for simulating the diffusive transport of water in the soil (Měch and Prusinkiewicz, 1996). A library of communication functions and the C source code of sample programs also make it possible for the user to create new, more specialized environmental programs. Their specification in a general-purpose language such as C or C++ is admittedly less convenient than the definition of plant models in an L-system-based modeling language, but no special-purpose high-level language for defining the multitude of possible environmental processes currently exists.

The Browser

In modeling practice, one usually deals with several models at once. Typically they are a mixture of models developed previously, subsequent refinements or alternative versions of the model currently being developed, and sample programs serving as examples of L-system code. L-studio and Vlab include a specialized graphical browser for organizing and accessing such models (Federl and Prusinkiewicz, 1999; Figure 7). A database of models is visualized as a hierarchical structure, with links relating older models to their refinements. Each model is associated with an icon, which facilitates its recognition. The browser makes it possible to access models on both local and remote machines. The remote access feature is particularly useful when a model is developed as part of a collaborative research project involving different sites.

Sample Models

L-studio and vlab are distributed with a set of sample and educational models, which illustrate various features of cpfg, its application areas, and modeling styles. A smaller set of lpfg models illustrates the main differences between the cpfg and lpfg modeling languages. The cpfg models are grouped into the following classes: a) models illustrating basic L-system constructs; b) developmental models controlled by lineage; c) developmental models controlled by endogenous signals and the allocation of resources; d) models simulating the interaction of plants with the environment; e) individual-based models of plant ecosystems; f) descriptive models of plant structures; g) models illustrating advanced modeling and visualization techniques; and h) models of mathematical concepts and physical objects other than plants (e.g., fractals, sea shells, molecules, and reaction-diffusion patterns).

APPLICATIONS

The interdisciplinary character of plant modeling research is echoed by the diversity of the existing and prospective applications of the models. From a horticultural perspective, the models aim at increasing our fundamental knowledge of plant biology and applying this knowledge to plant management and use, for example in computer-assisted landscape and garden designs. A list of existing and prospective applications of virtual plants has been compiled by Room et al. (1996). Its updated version, emphasizing horticultural applications, is given below.

- A. Advancement of science:
 - Plant biology: Exploration of hypotheses relating genetics, physiology and biomechanics to the development of integrated plant structures. Classification of plant architectures.
 - Ecology: Exploration of plant plasticity, including competition for space and resources at the level of single shoots, roots, and individual plants. Exploration of the relation between plant architecture and flower pollination by wind or insects. Improved understanding of insect behavior through simulation of insect movement and feeding on growing plants.
 - Evolution: analysis of plant design from an evolutionary perspective.
- B. Construction of practical models, in which plant architecture is of essence:
 - Plant breeding and genetic engineering: Specification of target “designer plants” by identification of architectures optimal for interception of light, harvestability, damage compensation, aesthetic appeal, quality and quantity of fruit, quality and timing of flower production, etc.
 - Plant treatment: Identification of planting strategies (spacing) and optimal treatments (pruning and pinching, training, application of chemicals, temperature and day length manipulation), for orchard, greenhouse, garden, and house plants.
 - Visual presentation: Creation of 3D models of ornamental plants for inclusion in plant registries, and for advertising plant varieties to customers.
 - Landscape architecture: Simulations of interactions between trees and structures; pruning strategies to minimize tree contact with power lines; interplanting for continuous flower displays.
- C. Construction of practical models, in which architecture is not of primary importance, but the inclusion of architecture may improve the predictive value of the models:
 - Management of pathogens: Improved understanding of disease dynamics through simulation of pathogen deposition and growth in the microclimates produced by developing plants.
 - Biological control of weeds: Identification of combinations of weed architecture and types of damage caused by herbivores or pathogens which interact particularly effectively to limit weed populations.
 - Management of pests: Improved definition of action thresholds through the simulation of interactions between plant architecture, pesticide deposition, insect movement and feeding, and compensatory growth of plants.
- D. Educational applications: Explanation and illustration of processes in plants, and the relationships between plants and their environment, for teaching and training purposes.

While the computer science techniques involved in the specification and visualization of the models seem to be relatively mature, the development of well calibrated empirical models of specific plants remains a labor-intensive task, and the construction of faithful mechanistic models is a current research problem.

Selected examples of virtual plants created using L-studio and Vlab are shown in Figures 8 to 14.

CONCLUSIONS

Many virtual plants have already been built. Some of these models have contributed to plant biology, by capturing and explaining selected mechanisms of the development of plant form. Nevertheless, practical applications of virtual plants to horticulture, agriculture, and forestry are still in their infancy. This appears to be the combined result of several factors:

- Physiological mechanisms underlying plant functioning are not known in sufficient detail to provide a firm basis for the construction of predictive models. Even less is known about the genetic basis of the development of whole plant forms.
- Plant modeling methodology and the development of the corresponding modeling

software are areas of active research. It takes time to apply the results to practical plant science problems.

- Construction of plant models is an interdisciplinary endeavor, which combines aspects of plant biology, computer science, and specialized modeling skills. It is difficult to find people with this combination of skills, or create teams that collectively have the required expertise.
- Construction of well calibrated models requires extensive measurements of plants and, therefore, is expensive.

This list suggests a direction for future organizational efforts (e.g., the multi-disciplinary training of prospective plant modelers) and for further research. One exciting open problem is the incorporation of genetic mechanisms into virtual plants. This could provide new insights into the fundamental mechanisms of development, from the level of genes to the final forms of plants around us, and further advance our understanding of nature through the synergy between biology, computer science, and art.

ACKNOWLEDGEMENTS

I would like to thank Drs. Campbell Davidson and Tom Fernandez for their invitation for me to present the results at the IHC 2002, and encouragement to write this review, and Lynn Mercer for editorial help. The surveyed results were obtained with financial support from the Natural Sciences and Engineering Research Council of Canada, Human Frontier Science Program, and France-Canada Research Foundation.

Literature Cited

- Bell, A. 1991. Plant Form: An Illustrated Guide to Flowering Plants. Oxford University Press, Oxford.
- Boudon, F., Prusinkiewicz, P., Federl, P., Godin, C. and Karwowski, R. 2003. Interactive design of bonsai tree models. Computer Graphics Forum 22(3):591-599 (Proceedings of Eurographics 2003).
- Erstad, K.A. 2002. L-systems, Twining Plants, Lisp. Cand. Scient. Thesis, University of Bergen.
- Federl, P. and Prusinkiewicz, P. 1999. Virtual Laboratory: An interactive software environment for computer graphics. Proc. Computer Graphics International 1999. Canmore, Canada 7-11 June. p. 93-100 and 242.
- Ferraro, P., Godin, C. and Prusinkiewicz, P. 2002. L-systems and MTGs: Integrating simulation and formal analysis of architectural plant models. Proceedings of the 16th European Simulation Multiconference. Darmstadt, Germany 3-5 June. p. 418-422.
- Foley, J.D., van Dam, A., Feiner, S. and Hughes, J. 1990. Computer Graphics: Principles and Practice. Addison-Wesley, Reading.
- Gautier, H., M  ch, R., Prusinkiewicz, P. and Varlet-Grancher, C. 2000. 3D architectural modeling of aerial photomorphogenesis in white clover (*Trifolium repens* L.) using L-systems. Annals of Botany 85:359-370.
- Godin, C., Gu  don, Y., Costes, E. and Caraglio, Y. 1997. Measuring and analysing plants with the AMAPmod software. p. 53-84. In: M.T. Michalewicz (ed.), Plants to Ecosystems. Advances in Computational Life Sciences I. CSIRO Publishing, Melbourne.
- Hall  , F., Oldeman, R.A.A. and Tomlinson, P.B. 1978. Tropical Trees and Forests: An Architectural Analysis. Springer-Verlag, Berlin.
- Harper, J.L. 1977. Population Biology of Plants. Academic Press, London.
- Herman, G.T. and Rozenberg, G. 1975. Developmental Systems and Languages. North-Holland, Amsterdam.
- Jirasek, C., Prusinkiewicz, P. and Moulia, B. 2000. Integrating biomechanics into developmental plant models expressed using L-systems. Plant Biomechanics 2000. Proceedings of the 3rd Plant Biomechanics Conference. Freiburg-Badenweiler, Germany 27 August – 2 September. p. 615-624.
- Karwowski, R. and Prusinkiewicz, P. 2003. Design and implementation of the L+C

- modeling language. Electronic Notes in Theoretical Computer Science 86.2.
- Lindenmayer, A. 1968. Mathematical models for cellular interaction in development, Parts I and II. Journal of Theoretical Biology 18:280-315.
- Měch, R. and Prusinkiewicz, P. 1996. Visual models of plants interacting with their environment. Proc. SIGGRAPH 1996. New Orleans, LA 4-9 August. p. 397-410.
- Mercer, L., Prusinkiewicz, P. and Hanan, J. 1990. The concept and design of a virtual laboratory. Proc. Graphics Interface '90. Halifax, Canada 14-18 May. p. 149-155.
- Petzold, C. 1998. Programming Windows. Microsoft Press, Redmont
- Prusinkiewicz, P., James, M. and Měch, R. 1994. Synthetic topiary. Proc. SIGGRAPH 1994. Orlando, FL 24-29 July. p. 351-358.
- Prusinkiewicz, P., Hammel, M., and Měch, R. 1997. Visual models of plant development. p. 535-597. In: G. Rozenberg and A. Salomaa (eds.), Handbook of Formal Languages, Vol. 3. Springer, Berlin.
- Prusinkiewicz, P. 1998a. In search of the right abstraction: the synergy between art, science, and information technology in the modeling of natural phenomena. p. 60-68. In: C. Sommerer and L. Mignonneau (eds.), Art at Science. Springer, Wien,.
- Prusinkiewicz, P. 1998b. Modeling of spatial structure and development of plants: a review. Scientia Horticulturae 74:113-149.
- Prusinkiewicz, P. 1999. A look at the visual modeling of plants using L-systems, Agronomie 29:211-224.
- Prusinkiewicz, P., Hanan, J. and Měch, R. 2000a. An L-system-based plant modeling language. p. 395-410. In: Nagl, M., Schürr, A. and Münch, M. (eds.), Applications of Graph Transformations with Industrial Relevance. Lecture Notes in Computer Science 1779. Springer, Berlin.
- Prusinkiewicz, P., Karwowski, R., Měch, R. and Hanan, J. 2000b. L-studio/cpfg: A software system for modeling plants. p. 457-464. In: Nagl, M., Schürr, A. and Münch, M. (eds.), Applications of Graph Transformations with Industrial Relevance. Lecture Notes in Computer Science 1779. Springer, Berlin.
- Prusinkiewicz, P., Mündermann, L., Karwowski, R. and Lane, B. 2001. The use of positional information in the modeling of plants. Proc. SIGGRAPH 2001. Los Angeles, CA 12-17 August. p. 289-300.
- Ratcliffe, O.J., Amaya, I., Vincent, C.A., Rothstein, S., Carpenter, R., Coen, E.S. and Bradley, D.J. 1998. A common mechanism controls the life cycle and architecture of plants. Development 125:1609-1615.
- Ratcliffe, O.J., Bradley, D.J. and Coen, E.S. 1999. Separation of shoot and floral identity in *Arabidopsis*. Development 126:1109-1120.
- Room, P.M., Hanan, J.S. and Prusinkiewicz, P. 1996. Virtual plants: new perspectives for ecologists, pathologists, and agricultural scientists. Trends in Plant Science, 1:33-38.
- Thornley, J.H.M. and Johnson, I.R. 1990. Plant and Crop Modeling: A Mathematical Approach to Plant and Crop Physiology, Oxford University Press, New York.
- da Vinci, L. 1970. The Notebooks of Leonardo da Vinci, Compiled and Edited from the Original Manuscripts by Jean Paul Richter. Dover Publications, New York.
- Woo, M., Neider, J., Davis, T. and D. Shreiner, D. 1999. OpenGL 1.2 Programming Guide, Third Edition. The Official Guide to Learning OpenGL, Version 1.2. Addison-Wesley, Reading.

Figures

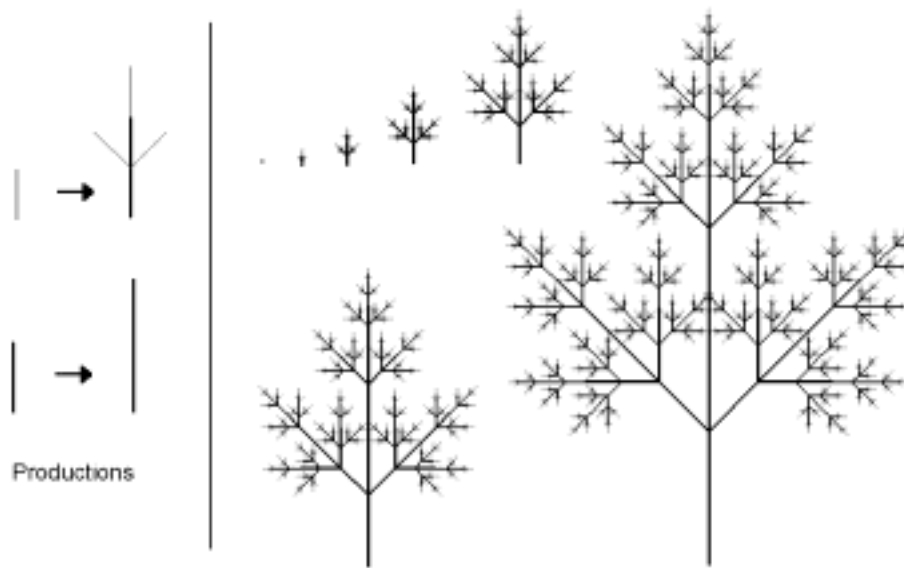


Fig. 1. Developmental model of a stylized compound leaf, viewed as a configuration of apices (light lines) and internodes (dark lines). From Prusinkiewicz et al., 1997.

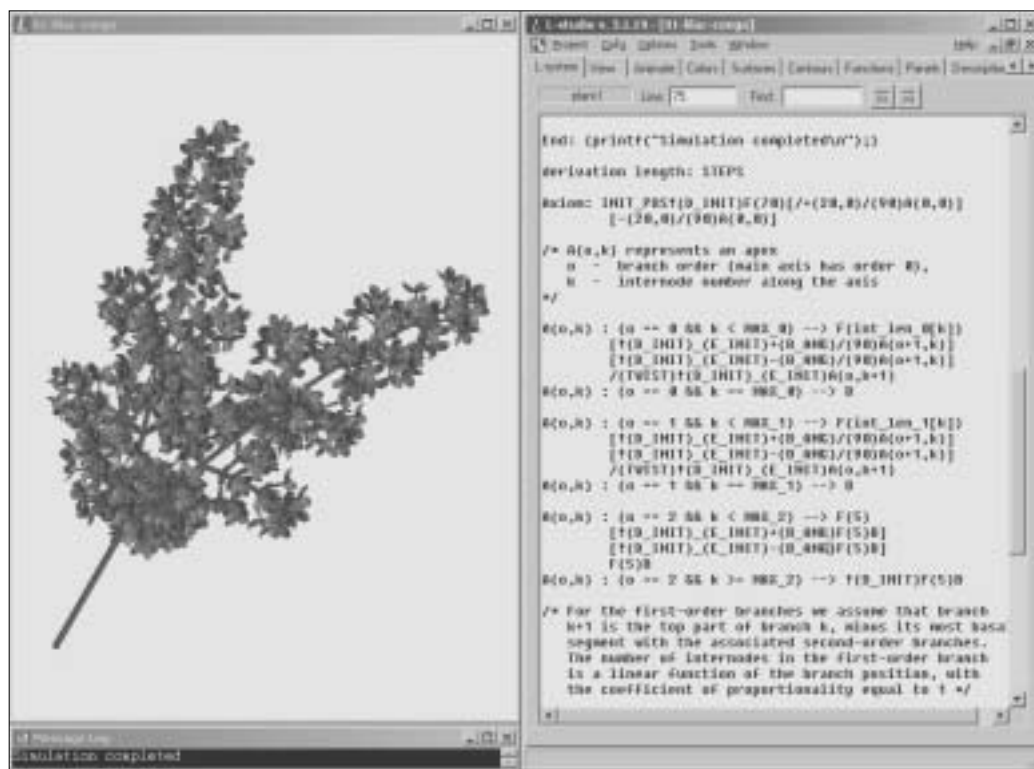


Fig. 2. Snapshot of the L-studio screen. A plant model visualized by cpfg is displayed on the left side of the screen. An auxiliary window underneath displays output and error messages. The window to the right provides access to various editors. In this example, a text editor is open on the file that specifies the main characteristics of the model in the cpfg modeling language. From Prusinkiewicz et al., 2000b.

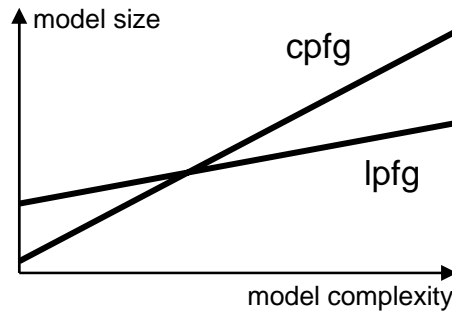


Fig. 3. The size of a model (in lines of code) as a function of its complexity. Cpfg is advantageous for simple models, while lpfg is advantageous for more complex models.



Fig. 4. Snapshots from an L-system-based simulation. Consecutive frames of this type result in the animation of development. From Prusinkiewicz et al., 2000b.

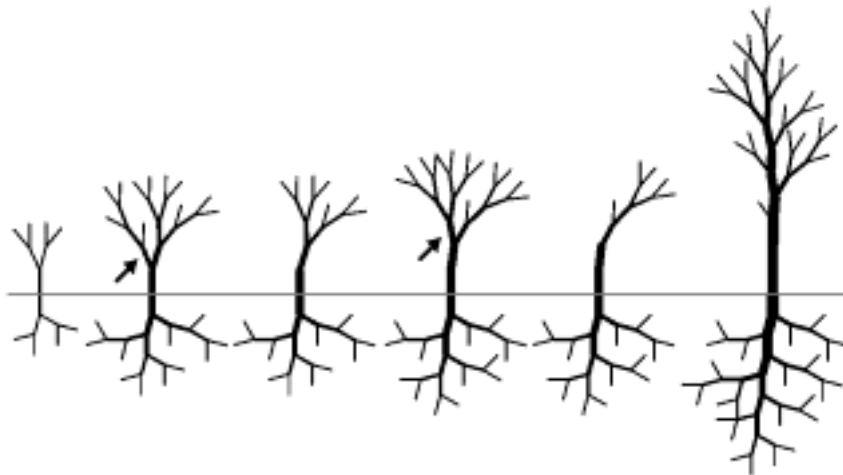


Fig. 5. Interactive manipulation of a schematic model simulating the maintenance of the shoot-root ratio. At chosen stages of plant development, the user has interactively removed a branch indicated by an arrow. The model has reacted by slowing down the growth of the root, and promoting new growth in the crown.

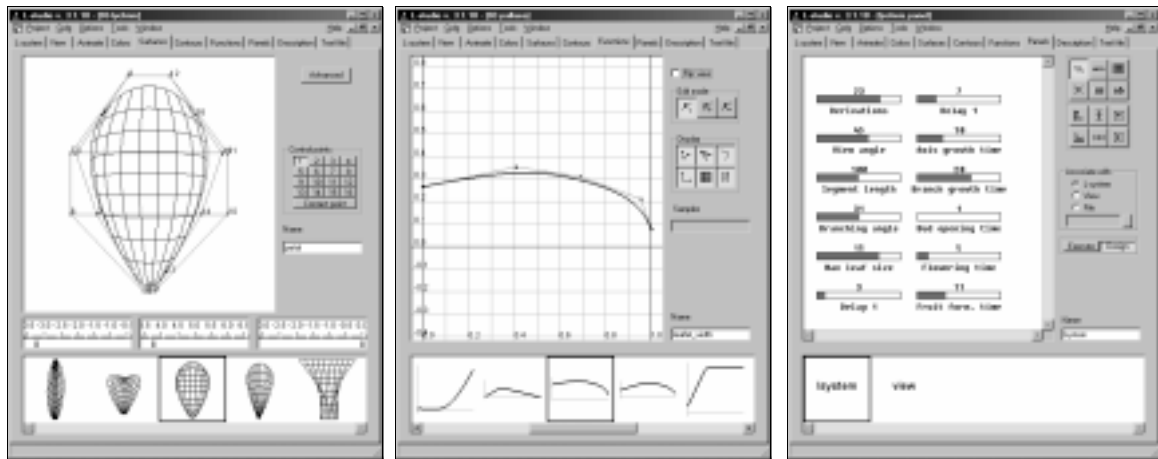


Fig. 6. Sample L-studio editors: (a) the surface editor, (b) the function editor, (c) a user-configurable control panel. The windows at the bottom of each editor are galleries, which make it possible to select a model component to be edited.

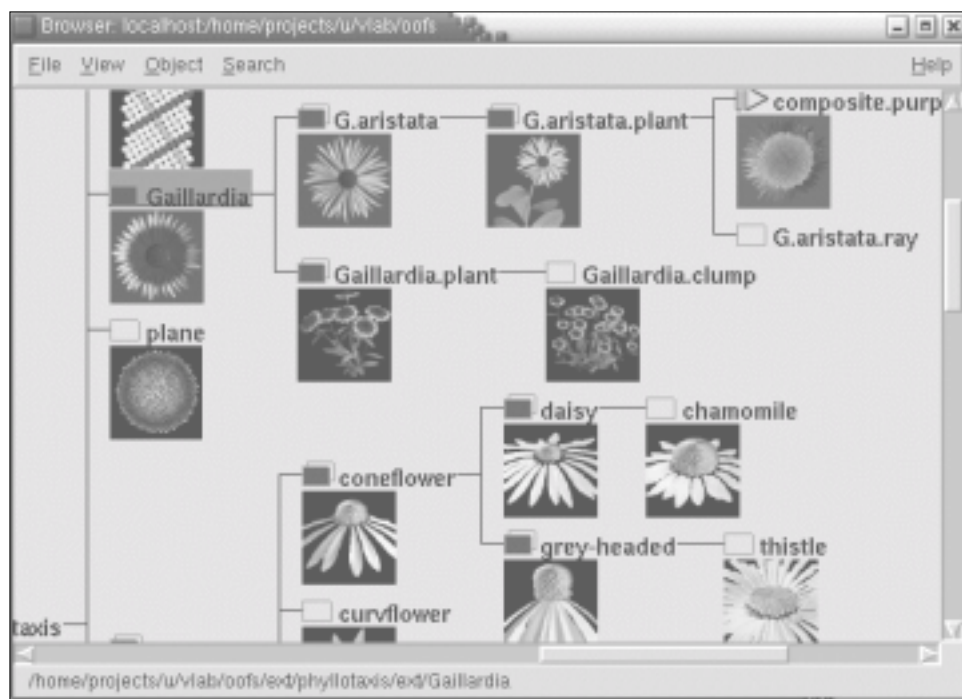


Fig. 7. A view of the L-studio/vlab browser. The browser provides easy access to databases of models, which may be stored on the local or remote machines.

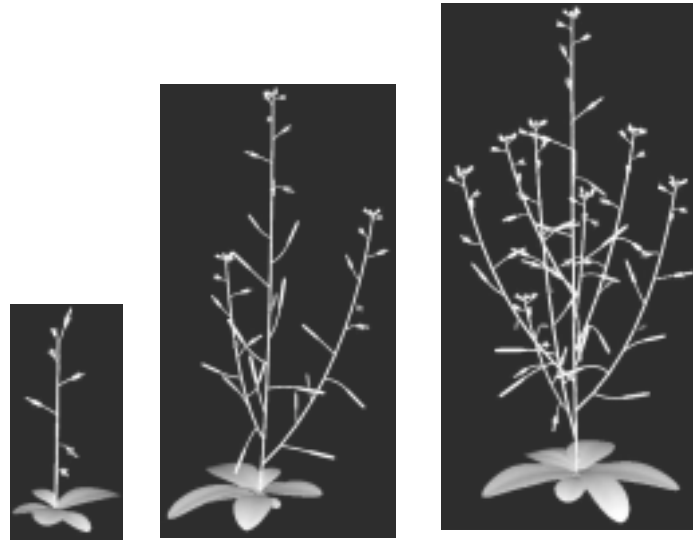


Fig. 8. Models of mutant and transgenic *Arabidopsis thaliana* plants. The models illustrate the emerging potential of relating plant appearance to the genetic makeup of the plant. From left to right: the *tfl1* mutant, wild type, and the 35*STFL1*;35*SLFY* transgenic plant. E. Coen, K. Lee, L. Mündermann, and P. Prusinkiewicz, 2002 (unpublished), based on Ratcliffe et al., 1998, 1999.



Fig. 9. A physiologically-based model of white clover (*Trifolium repens* L.) response to local light. Left: model grown in full light ($\text{PAR} = 500 \mu\text{mol m}^{-2} \text{s}^{-1}$ and $\text{R/FR} = 1.15$). Right: model grown in green shade ($\text{PAR} = 150 \mu\text{mol m}^{-2} \text{s}^{-1}$ and $\text{R/FR} = 0.1$). From Gautier et al., 2000.

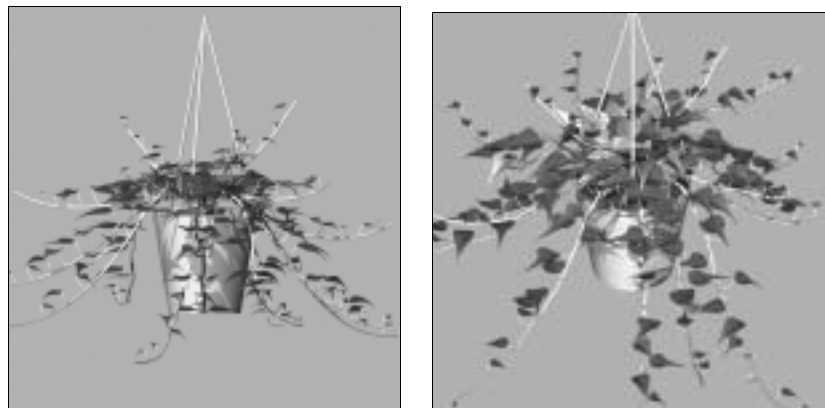


Fig. 10. Two views of a biomechanical model of a hanging plant. The model captures the combined effect of gravity and tropism on the shape of branches of a growing plant. From Jirasek et al., 2000.

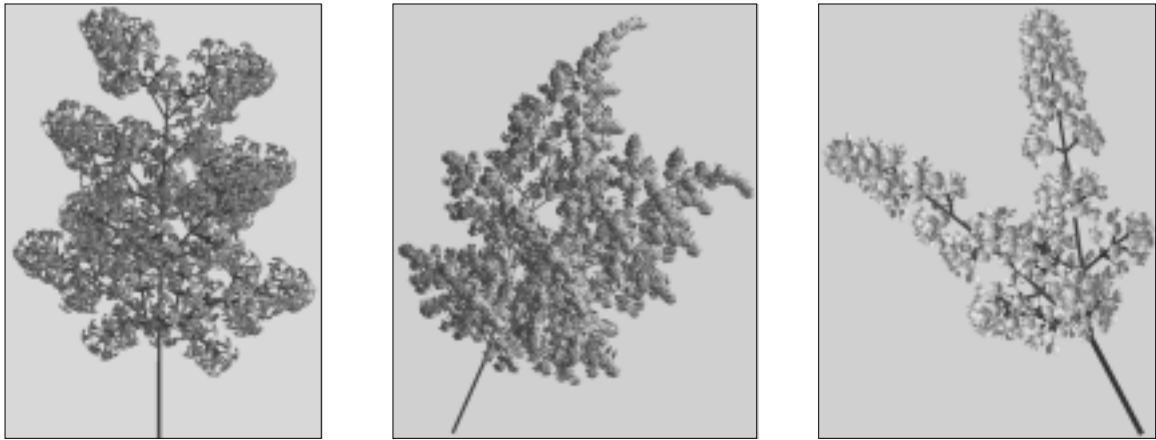


Fig. 11. Three-dimensional descriptive models of lilac inflorescences: *Syringa chinensis* 'Rubra', *Syringa reticulata*, *Syringa vulgaris* 'Prinzessin Klotilde'. Such models could be used in plant registries and in advertising of plants. C. Davidson and P. Prusinkiewicz, 1993 (unpublished).



Fig. 12. Bonsai tree models, created using models that respond to a variety of operations including pruning & bending of branches. From left to right: chokkan (formal upright), bunjinji, and sankan (triple trunk) style. From Boudon et al., 2003.



Fig. 13. Topiary garden models, which incorporate trees and shrubs that respond to pruning. From Prusinkiewicz et al., 1994; and Prusinkiewicz, 1998b.



Fig. 14. A landscape with trees modeled using L-systems. From Prusinkiewicz et al., 2001.