# The L−system−based plant−modeling environment L−studio 4.0

Radoslaw Karwowski and Przemyslaw Prusinkiewicz
Department of Computer Science, University of Calgary

## Introduction

L−studio and the Virtual Laboratory (Vlab) are two related plant modeling packages developed at the University of Calgary, Canada. They have been distributed by this University since 1999. The packages run on Windows and Linux machines, respectively, 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. Below we outline the most recent version of L−studio (4.0).

## Reference

Radoslaw Karwowski and Przemyslaw Prusinkiewicz. The L−system−based plant−modeling environment L−studion 4.0. In *Proceedings of the 4th International Workshop on Functional−Structural Plant Models*, pp. 403−405.

# The L-system-based plant-modeling environment L-studio 4.0

Radoslaw Karwowski and Przemyslaw Prusinkiewicz
Department of Computer Science, University of Calgary
Calgary, Alberta, Canada T2N 1N4
{radekk|pwp}@cpsc.ucalgary.ca

## Introduction

L-studio and the Virtual Laboratory (Vlab) are two related plant modeling packages developed at the University of Calgary, Canada. They have been distributed by this University since 1999. The packages run on Windows and Linux machines, respectively, 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. Below we outline the most recent version of L-studio (4.0). Additional information and applications are presented in (Prusinkiewicz 2004). A user's guide is available online (Karwowski and Lane, 2004).

The L-studio 4.0 package consists of: (a) two L-system-based simulation programs, cpfg and lpfg; (b) a modeling environment that provides auxiliary modeling tools; (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.

## Simulation programs: cpfg and lpfg

The simulation programs cpfg and lpfg are at the heart of L-studio. 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 users in L-system-based languages. The modeling language used in cpfg, developed first, is based on the mathematical notation of L-systems, and makes it possible to specify simple models very concisely. The modeling language used in lpfg, called L+C, incorporates the L-system notion of productions into the full implementation of the C++ programming language (Karwowski 2002; Karwowski and Prusinkiewicz; 2003, see also Prusinkiewicz et al., 1999). 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 several alternative successors, and a mechanism for fast transfer of unidirectional information. This last feature makes it possible to solve systems of differential equations defined on the simulated structure (Federl and Prusinkiewicz, 2004). Such equations occur, for example, when describing transport and partitioning of resources in a plant, or calculating the mechanical equilibrium state of biomechanical models (Jirasek et al., 2000). As a result, L+C is advantageous when developing complex plant models.

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 to trigger a programmable action associated with this component. Some applications of this feature include: simulation of pruning and grafting, 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.

Based on their input (a number of files including the L-system model and optional data files), cpfg and lpfg create a three-dimensional internal representation of a model and project it on the screen. Model visualization is based on the OpenGL graphics library and supports standard modeling and rendering techniques developed in computer graphics, such as parametric surfaces, generalized cylinders, and texture mapping. The output may be a static model (which can be interactively rotated and zoomed in/out by the user) or computer-generated animation that results from visualizing consecutive stages of the simulation. The visualizations may have the form of schematic or realistic representations of branching structures, as well as plots and histograms. The multi-view presentation mode, available in lpfg, makes it possible to simultaneously visualize different aspects of the simulation in separate windows on the screen. For instance, one view may realistically depict a growing plant, while another view may represent quantitative aspects of the simulation in progress as function plots.

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 more specialized programs, such as AMAPmod, a program dedicated to analyzing plant architectures (Godin et al., 1997; Ferraro et al., 2002).

### L-studio editors

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 provides a user interface for specifying this complementary information and transferring it to the simulation programs. The L-studio graphical interface (Prusinkiewicz et al., 2000) 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. The available editors include: a *text* editor for creating and manipulating L-system and other textural files that constitute the model; a *control panel* manager, which makes it possible to create user-configurable controls, such as sliders and buttons, for manipulating model parameters; *surface* and *cross-section* editors for defining the shape of individual plant organs; *color* and *material* editors for specifying plant appearance when rendering; and a graphical editor of *functions* of a single variable. 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 makes it possible to simulate such interaction using concurrently running, communicating programs that represent the plant and its environment (Měch and Prusinkiewicz, 1996). The available environmental programs include: programs that model collisions between plant organs and their environment, programs for simulating the diffusive transport of water in the soil, and several programs for calculating the distribution of light in a plant canopy. 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

During the modeling process, 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 4.0 includes a specialized graphical browser for organizing and accessing such models, which is based on an earlier Vlab implementation (Federl and Prusinkiewicz, 1999). A database of models is visualized as a hierarchical structure, 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 is distributed with a set of sample and educational models, which illustrate various features of cpfg and lpfg, their application areas, and modeling styles. 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). The lpfg versions of many of these models are also available.

## Acknowledgments

This research has been supported by the user community of L-studio and by grants from the Natural Sciences and Engineering Research Council of Canada and the Human Frontier Science Project.

## References

Federl, P. and Prusinkiewicz, P. 1999. Virtual Laboratory: An interactive software environment for computer graphics. Proceedings of Computer Graphics International 1999, Canmore, Canada, pp. 93-100 and 242.

Federl, P. and Prusinkiewicz, P. 2004. Solving differential equations in developmental models of multicellular structures using L-systems. To appear in Lecture Notes in Computer Science 3037 (Proceedings of the International Conference in Computational Science ICCS 2004, Krakow, Poland, Part II), pp. 67-74.

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, pp. 418-422.

Godin, C. Guédon, Y., Costes, E. and Caraglio, Y. 1997. Measuring and analysing plants with the AMAPmod software. In: M.T. Michalewicz (ed.), Plants to Ecosystems. Advances in Computational Life Sciences I. CSIRO Publishing, Melbourne, pp. 53-84.

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, pp. 615-624.

Karwowski, R. 2002. Improving the Process of Plant Modeling: The L+C Modeling Language. Ph.D. thesis, University of Calgary.

Karwowski, R. and Prusinkiewicz, P. 2003. Design and implementation of the L+C modeling language. Electronic Notes in Theoretical Computer Science 86(2).

Karwowski, R. and Lane, B. (2004) L-studio 4.0 user's guide. http://www.cpsc.ucalgary.ca/Research/bmv/lstudio.

Měch, R. and Prusinkiewicz, P. 1996. Visual models of plants interacting with their environment. Proceedings of SIGGRAPH 1996. New Orleans, LA, pp. 397-410.

Petzold, C. 1998. Programming Windows. Microsoft Press, Redmond.

Prusinkiewicz, P., Karwowski, R., Perttunen, J., and Sievänen, R. (1999). Specification of L – a plant-modeling language. Manuscript, University of Calgary.

Prusinkiewicz, P., Karwowski, R., Měch, R. and Hanan, J. 2000. L-studio/cpfg: A software system for modeling plants. In: Nagl, M., Schürr, A. and Münch, M. (eds.), Applications of Graph Transformations with Industrial Relevance. Lecture Notes in Computer Science 1779. Springer, Berlin, pp. 457-464.

Prusinkiewicz, P. 2004. Art and Science for Life: Designing and Growing Virtual Plants with L-systems. Acta Horticulturae (ISHS) 630, 2004, pp. 15-28.