Manipulating virtual plants

Przemyslaw Prusinkiewicz[1], Brendan Lane[1], and Radomir Mech[2]
[1]Department of Computer Science, University of Calgary, Canada
[2]Adobe, San Jose, CA

Keywords: L-system, simulation, plant-modeling software design, virtual laboratory, L-studio, model specification and exploration, human-computer interaction

Summary: We propose programming constructs for interacting with virtual plants modeled using L-systems, and survey application areas that may benefit from interactive simulations.

Introduction and previous work

The first three-dimensional plant simulation models were implemented over 35 years ago [Honda 1971]. Since then, processor speed has been increasing exponentially (Moore's law), and powerful graphics cards have become ubiquitous. These technological advancements have had a significant impact on the methods for simulating and visualizing plants. The complexity of plant models has increased, and simulations that once could only be appreciated using single-frame animation (recording individual stages of the simulation, than playing them back as a movie) can now be viewed in real time, as the simulation progresses.

Fast simulations set the stage for interaction with plant models. First, graphically controlled sliders were introduced for manipulating model parameters in real time [Oppenheimer, 1986]. This concept was generalized into panels, which the user could configure and apply to control arbitrary model parameters. [Mercer et al 1990, Prusinkiewicz and Lindenmayer 1990]. Further extensions included graphically-defined functions, which made it possible to interactively specify and modify relations between parameter values [Lintermann and Deussen 1999, Prusinkiewicz 2004]. Plant models represented on the screen also lend themselves to direct manipulation [Shneiderman 1987], creating the impression of interacting with the plants themselves. Previous work in this area includes interactive bending and pruning of branches [Power et al. 1999, Boudon et al. 2003], and sketch-based techniques [Ijiri et al. 2005] for interactively specifying plant geometry.

Our work is focused on interactions with complex functional-structural plant models using visual techniques that resemble manipulations of real plants. For example, selected branches of a virtual tree trained for an espalier may be bent and attached to wires. This will affect the development and growth of further branches, which may be subject of manipulation at a later date. Creation of virtual plants capable of such interactions represents a methodologically challenging problem due to the contrasting demands of simulations and interactive control. On one hand, plant structure and behavior are defined by program execution; on the other, they are subject to user actions. Furthermore, the responses to user actions should themselves be programmable, to allow for easy changes and refinement during model development.

Implementation

We present a solution of this problem implemented in the L-system-based simulation program lpfg, a component of the L-studio and Virtual Laboratory modeling software [Prusinkiewicz

2004]. At the base of our implementation are two standard low-level operations of human-computer interface design: selection of a component, and location of a point on the screen [Foley et al. 1996]. In lpfg, the user distinguishes between these operations by pressing appropriate control keys on the keyboard. When a component of a virtual plant is *selected* with the mouse, a predefined module `MouseIns()` is inserted before the selected component's representation in the L-system string. When a point is *located* with the mouse, its coordinates are returned to the L-system-based model by calling the predefined function `GetMouseStatus()`. This function returns screen coordinates of the located point as well as object-space coordinates of the line traced from the located point into the screen. The automatic transformation of the traced line to the object-space coordinates makes it easy for the modeler to relate mouse position to the model using the coordinates in which the model is expressed.

The `MouseIns()` and `GetMouseStatus()` constructs are powerful building blocks for creating interactive virtual plants, and are well integrated into the L-system formalism. The module `MouseIns()` can appear on par with other L-system modules as a predecessor or context of productions. Thus, the designer of an L-system model can specify the reaction to module selection using the same formalism of L-system productions as that used to specify non-interactive aspects of the model behavior. Furthermore, in many applications the selection and location operations can be seamlessly combined, allowing for three-dimensional manipulations using a two-dimensional input device, a mouse. For example, a branch can be bent to a new position by selecting a branch component, then dragging it in the plane parallel to the screen with the mouse. Two coordinates of the new location are then calculated on the basis of the position of the mouse-controlled cursor, while the third coordinate, representing distance from the screen, is derived from the original depth of the selected component.

Applications

Although module selection has been available within the L-studio for some time, the recent addition of the location capability greatly enhances the spectrum of possible interaction modes. We are exploring the following application domains:

Horticultural plant manipulation. For example, a carbon-allocation model of a fruit tree may allow the user to pick selected fruits from a virtual tree, simulating thinning. A biomechanical tree model intended for designing topiary gardens, espaliers, or bonsai may allow the user to virtually prune branches, bend them, and tie them to wires. A plant model may also support virtual grafting operations.

Laboratory experiments and simulation of plant responses to stimuli. A virtual plant designed to study hormonal control of plant development may support interactive application of hormones to selected plant parts. The user may interactively deposit insects or pathogens on selected plant organs, inducing simulated defense mechanisms. The user may also "touch" the plant, inducing responses such as the collapse of Mimosa leaves, or the folding-up of the Venus flytrap [Simons 1992].

Interactive visualization. Examples include:

- Attachment of "probes", which dynamically monitor the state of selected modules of a model. The results may be presented in textual or graphical (histogram) form near the selected modules or in a separate window, and are useful both for monitoring simulated experiments and when debugging and testing plant models.

- Display of background reference information related to the selected modules or plant components. For example, selection of a flower in the model may create a new window which displays the corresponding image of the real flower, animation of its development, or references to the experimental data and literature used to construct the model.
- Modification of plant geometry, as needed to reveal hidden components. For example, leaves hiding the meristem from view can be removed (as is often done in microscopy), or internodes between rosette leaves can be made longer to expose lateral buds and the sequence of their development.
- Attachment of a virtual camera to specific parts of the plant. This can be used to focus on the development of a specific plant component, such as a leaf, flower, or shoot apex.

Conclusions

The augmentation of the L-system-based modeling language L+C with simple programming constructs for selecting modules and locating points on the screen has opened the door for constructing a wide range of interactive plant models. Control panels remain useful for specifying global parameters of the models, while direct manipulation makes it possible to interact with a virtual plant locally and in a more intuitive manner, resembling the manipulation of real plants. This is useful while experimenting with the models for research purposes, and has particular appeal in models designed for education and training.

The models outlined in this abstract represent only a small fraction of the wide range of possible interactive plant models. Construction of further models is an exciting area of research. As interactive models become more numerous and diverse, it will become increasingly important to inform the end-user of the operations that can be performed, and their meaning in the context of a specific model. What to select? Where to move? How to interpret the output? An instruction manual accompanying a model is a possible but unattractive solution. Self-documenting models, providing hints when needed, are a more appealing alternative. Concepts and programming constructs for creating such models are also an area of further research. Finally, on the more technical end of the spectrum of open problems is support for recording and "playing back" interactive simulations, so that the interwoven track of user manipulations and simulations can be easily reproduced.

Acknowledgments

The idea of accessing background information through the model was originally proposed by Andrew Bangham. The notion of probes was proposed by Mitch Allen. The initial implementation of module selection in the context of L-system was by Radoslaw Karwowski. Support of this research by the Natural Sciences and Engineering Research Council of Canada and Adobe is gratefully acknowledged.

References

F. Boudon, P. Prusinkiewicz, P. Federl, C. Godin and R. Karwowski [2003]. Interactive design of bonsai tree models. Proceedings of Eurographics 2003: *Computer Graphics Forum* 22 (3) (Proceedings of Eurographics 2003), pp. 591-599.

J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes [1996]: *Computer Graphics. Principles and Practice*. Addison-Wesley, Reading.

H. Honda [1971]: 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 Theoretiocal Biology* 31:331-338.

T. Ijiri, S. Owada, M. Okabe, and T. Igarashi [2005]: Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints. *ACM Transactions on Graphics* 24 (3) (Proceedings of SIGGRAPH 2005), pp. 720-726.

B. Lintermann and O. Deussen [1999]: Interactive modeling of plants. *IEEE Computer Graphics and Applications* 19(1):56-65.

L. Mercer, P. Prusinkiewicz, J. Hanan [1990]: The concept and design of a virtual laboratory. Proceedings of Graphics Interface ’90, pp. 149-155.

P. E. Oppenheimer [1986]: Real-time design and animation of fractal plants and trees. *Computer Graphics* 20 (4) (Proceedings of SIGGRAPH ’86), pp. 55-64.

J. L. Power, A. J. Bernheim Brush, P. Prusinkeiwicz, and D. H. Salesin [1999]. Interactive arrangement of botanical L-system models. Proceedings of the 1999 Symposium on Interactive 3D Graphics, pp. 175-182 and 234.

P. Prusinkiewicz, A. Lindenmayer [1990]: *The Algorithmic Beauty of Plants*. Springer, New York. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer and L. Mercer.

P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane [2001]. The use of positional information in the modeling of plants. Proceedings of SIGGRAPH 2001, pp. 289-300.

P. Prusinkiewicz [2004]: Art and science for life: Designing and growing virtual plants with L−systems. *Acta Horticulturae* 630 (2004), pp. 15−28.

B. Shneiderman [1987]: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading.

P. Simons [1992]. *The Active Plant*. Blackwell, Oxford.