Computer Science Honours
Procedural Modelling of Plant Scenes

Author: Kim Randleff-Rasmussen

Supervisors: Shaun Bangay

                 Adele Lobb

Literature Survey

1. Introduction

This project will model a number of plant scenes using L-Systems and procedural modelling. The modelling of the plants to be rendered will be achieved through the use of L-Systems. These modelled plants will then be brought together into a single scene using procedural modelling.

The following literature review summarises the information available on L-Systems and other plant modelling techniques, procedural modelling and computer graphics software, with focus on those which concentrate on translation of L-Systems.

The overview is structured as follows: firstly, a description of L-Systems is given followed by an investigation into work that has been conducted on, with and concerning L-Systems. Section 2 looks at other ways of modelling plants and organic structures, including examples of the use of plant modelling techniques in practice. Section 3 explores procedural modelling, focussing on instances in which procedural modelling has been used to model geometrically complex scenes. Lastly, Section 4 contains a review of current graphics software which provides interpretation of L-Systems.

2. L-Systems

2.1. Definitions and Explanations

In 1968, Hungarian biologist Aristid Lindenmayer developed a formalism for multicellular development [Lindenmayer 1968]. This formalism, subsequently named L-Systems (Lindenmayer Systems), captured the imagination of computer graphics artists who extended and improved the basic string rewriting language to form a model. This model is now widely used in computer graphics for the modelling of plants and other organic structures.

An L-System is a string rewriting formalism which can be interpreted graphically by a turtle graphics program. These rewritings can be applied sequentially or in parallel, to every module or according to contextual information. There has even been the definition of a type of L-System which allows for both – the rewriting of two or more symbols with a single production [Prusinkiewicz 1986]. The rewritten strings are then parsed by a LOGO-style graphics turtle which interprets each symbol in the string as an action – for example, F often means draw a line forward and + represents a rotation around the y-axis by a number of degrees which is defined by a constant or a parameter [Szilard and Quinton 1979].

Since the original definition of L-Systems, many formal explanations and definitions of L-Systems have been published. Among the more prominent writers is Professor Przmyslaw Prusinkiewicz from the University of Calgary who, along with Lindenmayer, wrote what is considered to be the authoritative work on the subject [Prusinkiewicz and Lindenmayer 1990]. He has also written papers describing more specific types of L-Systems.

2.2. Extensions

Much has been published regarding different types of L-Systems which allow for certain changes in order to more accurately model reality. Rozenberg, for example, defined table L-Systems [Rozenberg 1973] which allow for the choosing of a set of developmental rules according to certain environmental factors. Environmentally sensitive L-Systems [Prusinkiewicz *et al.* 1994] were the next step, providing query symbols which allowed L-Systems to change as a result of local environmental conditions. Open L-Systems [Prusinkiewicz and Měch 1996] extended this functionality by including bilateral communication with the environment. This extension allowed the environment to be influenced by changes in the tree model, which enabled the realistic modelling of such aspects as nutrient uptake and supply.

Other extensions provide functionality for a single model to be varied and thus produce a number of individuals. These individuals share certain characteristics but are different and can thus be said to be individuals within a particular species or group. Parametric L-Systems [Prusinkiewicz *et al.* 1995] include the facility to involve parameters in a standard L-System which allows many aspects of the model to be changed dynamically rather than statically defined in the model. Such aspects as step length and rotation angle can easily be varied in this manner in response to environmental and local changes. Differential L-Systems (or dL-Systems) [Prusinkiewicz *et al.* 1993] extend these parametric L-Systems by incorporating continuous time flow. This facility allows smooth animation of plant development rather than the rendering of discrete development steps, as is the standard practice.

Perhaps the most widely used L-Systems in terms of individual variation are stochastic L-Systems [Eichhorst and Savitch 1980]. These use multiple productions for each symbol and assign probabilities to each production. The specific production which is chosen at run-time is then determined by a random number generator.

Many works exist which serve as extensions to Prusinkiewicz and Lindenmayer's original explanatory work. These compile all of the extensions made to L-Systems since 1990 into formal definitions and explanations including examples. A representative example of this synoptic work can be found in [Prusinkiewicz *et al.* 1996].

Some L-System extensions which are unlikely to be found in such formal definitions, are ones which are intended to model a specific element of plant development for a specific project. Some of these have been the inclusion of a cut symbol (%) [Hanan 1992] to model self-pruning (see also [Prusinkiewicz *et al.* 1994]) and the development of erasing productions [Herman and Rozenberg 1975] to enable individual variation in a standard model.

Prusinkiewicz and Lane define a particular type of L-System which allows a set of productions to apply to a set of strings (individual plants) instead of to a single string. These L-Systems, called multi-set L-Systems [Prusinkiewicz and Lane 2002], mean that a dynamic plant population can be maintained and operated on with a single

model. This technique was developed in order to manage spatial distribution of a plant population.

Much work has been done on the inclusion of genetic algorithms [McCormack 1993], mutation [Mock 1998] and evolution [Rodkaew *et al.* 2002] into L-Systems. This amalgamation of techniques allows for such applications as virtual landscaping and adaptation simulations. These simulations can be further aided by a technique known as extrusion in space-time [Hammel and Prusinkiewicz 1996] which extends a two-dimensional structure's development into a three-dimensional line or curve which represents the progress of time.

2.3. Graphical Interpretation

As mentioned, the most common way of translating L-Systems into three- or two-dimensional images is through turtle graphics. A reasonable summary of this technique is to be found in [Prusinkiewicz *et al.* 1995], an author of which also helped to develop a software program called CPFG [James *et al.* 1993] which puts these turtle interpretation principles into practice.

[Prusinkiewicz *et al.* 1999] describes some of the programming principles behind the graphical interpretation of L-Systems. This paper is particularly useful for readers looking for explanations of L-System translation from formal language to programming language.

3. Plant Modelling

3.1. Techniques

L-Systems are not the only plant modelling technique available. Many other models abound, some of which focus on plant geometry as L-Systems do, and others which model trees based on behavioural aspects.

Some well-known models in the former class include the one developed by de Reffye *et al.* [de Reffye *et al.* 1988] which models botanical rules in a mathematical model. Other models focus more on particular aspects of plant geometry. For example, Borchert and Honda [Borchert and Honda 1984] designed a model which illustrates the tendency of a tree to increase the vigour of growth after the loss of a branch. Meyerowitz [Meyerowitz 1994] used the specific plant *Aribidopsis thaliana* to model genes and mutation as they influence plant growth. Janssen and Lindenmayer looked at patterns of branching and flowering in *Mycelia muralis* (wall lettuce) [Janssen and Lindenmayer 1987].

A comprehensive study of plant architectures and models which reflect the range available can be found in [Hallé *et al.* 1978]. This book asserts that the thousands of plant species in the world can be modelled using a mere twenty-three models. These models take into account individual species architecture and genetic "growth programme" as well as competition for space and nutrients in a single forest. It goes one step further, too, and models the architecture of forests as separate entities. Borchert and Slade [Borchert and Slade 1981] also analysed the growth of a number of tree species in order to create models.

The second class of plant models focuses on how behaviour and the environment influence the manner in which a tree grows. [Prusinkiewicz *et al.* 2000], for example, models the effect of gravity on the branching behaviour of certain trees. Another paper by one of those authors looks at pruning – forced and self-imposed – and how this affects branch growth [Prusinkiewicz *et al.* 1994]. Yet another paper [Prusinkiewicz *et al.* 2001] uses positional information to model the growth and survival of a plant. This model uses such techniques as self-thinning and competition for space to model how tree growth is affected by environmental factors.

A model designed by Lück, Lück and Bakkali [Lück *et al.* 1990] suggests the use of a growth potential or vigour value to be assigned to branches. This value dictates which of a number of competing branches will triumph. Honda *et al.* also focussed on branch interaction between trees [Honda *et al.* 1981]. The value they use to determine the "winning" branch is flow rate which dictates the rate of bifurcation. Sorrensen-Cothern, Ford and Sprugel divide a single plant into a number of modules and model the interaction between these rather than interaction between entire plants [Sorrensen-Cothern *et al.* 1993].

Another trend to be found in plant modelling is the use of fractals as introduced by Mandelbrot in 1975. [Oppenheimer 1986] contains a useful model of plants through fractals as well as a summary of a number of tree models which have been published through the years. [Smith 1984] also summarises a number of plant models, both fractal and L-System based. His paper also contains a practical summary of formal languages as used in such modelling.

3.2.   Practical Applications

For completeness' sake, this section contains mention of a number of papers which describe the use of plant models (both L-System and other) in practice. [Mackrill 2003] used L-Systems to model realistic landscapes including trees, grass and terrain. His focus was on the ability to render in real-time. [Prusinkiewicz *et al.* 1988] describes the use of L-Systems to model not only branches, but leaves and flowers as well. There is focus, here, on the ability to show the animation of plant development, hence the choice of L-Systems as the modelling technique.

[Weber and Penn 1995] designed an entirely new plant modelling language which allowed for realistic depiction of geometry without strict adherence to botanical principles. They also presented a technique for varying detail level which eases the rendering of large, landscape shots of complex forests.

4.   Procedural Modelling

Procedural modelling is a computer graphics technique designed to decrease the computational power required to render a geometrically complex scene. Traditional rendering techniques retrieve each scene element from a database at run-time and insert them into the scene for rendering. This means that every element has to be rendered in its entirety for the scene to be rendered. Procedural modelling aims to solve this problem by only

including those objects that are visible in the scene [Foley 1996]. At run-time, a detailed model of the scene is built up with each scene element added if it is within the view spectrum of the camera.

Other techniques utilised in procedural modelling are the inclusion of elements according to the level of detail required. For example, a knot on a tree trunk would only be visible from a metre or less away from the tree and not from an aerial shot. Changing the detail of each element to be included is another way in which to decrease the computation required in rendering a scene.

A number of papers and books exists which explain these concepts behind procedural modelling. Some of these are [Jeschke *et al.* 2003] and [Macri and Pallister 2004]. Amburn *et al.* [Amburn *et al.* 1986] suggest a number of ways in which traditional procedural models might be improved. One of these techniques is communication between models which may affect one another that models may adjust their behaviour according to messages from another model. The second improvement is subdivision of labour and the sharing of some subdivided activities between procedures.

[Smith 1984] also discusses improvements to procedural models. His technique is one called "database amplification" and is a property of formal languages which allows the database of available symbols to be dynamically added to.

Procedural modelling has long been in use in programs which model geometrically complex scenes such as forests, cities and crowds. A number of papers present these programs, describing the use of procedural models to ease the rendering of complex scenes which contain many instances of geometrically similar constructs (such as trees and buildings). Some of the more prominent papers which model large cities are [Birch *et al.* 2003], [Greuter *et al.* 2003], [Parish and Müller 2001] and [Wonka *et al.* 2003]. In terms of forest or landscape generation, some papers to look at include [Guerraz *et al.* 2003] and [Deussen *et al.* 2002].

Many of such papers describe not only the use of procedural models, but also the interaction between the procedural models and other aspects of scene modelling. [Deussen *et al.* 2002] and [Greuter *et al.* 2003], for example, describe entire systems for scene modelling – drawing all aspects of modelling and rendering together.

5.   Software Products

Much research and development has been done into software programs that translate L-Systems into three-dimensional images. Other software products have also been developed which model plants and organic structures without the explicit use of L-Systems.

One of the first L-System translation systems was developed in 1970 [Baker and Herman 1970] in order to simulate organic development for hypothesis testing. Prusinkiewicz and his students and colleagues have developed a number of software products since that time. CPFG [Prusinkiewicz 1993] takes a parameterised procedural model of a plant and outputs a

geometric model. This product is used in the modelling system L-Studio [Prusinkiewicz 1998] which provides a graphical user interface for the modelling of plants via L-Systems with comprehensive options for all aspects of L-Systems.

Laurens Lapré, a student of Professor Prusinkiewicz, developed a system called Lparser [Lapré 1993] which is available in open source and is a good place to start for beginners in L-System translation due to the readable, logical code. Another product developed by a student, though not one taught directly by Prusinkiewicz, is L-Arbor developed by Marco Grubert [Grubert 1999]. This program offers simple animation and L-System modelling facilities and is an easy way to learn basic L-Systems.

The Institute of Forest Biometry and Informatics at the University of Göttingen is a special interest plant modelling group comprised of experts in a number of fields including forest science, biology and mathematics. Winfried Kurth is the computer graphics expert and he developed a modelling program called GROGRA (GROwth GRAmmar) [Kurth 1994].

GRAMPS [O'Donnell and Olson 1981] is a graphics language interpreter developed to model molecular structures and to animate their development. It is used in conjunction with an interactive vector display list processor and is focussed on the provision of real-time scene manipulation facilities. A nice feature of this program is the facility to dynamically add new graphics manipulation commands through the use of macros.

6.   Conclusion

L-Systems were chosen over any of the other available modelling techniques due to their pervasiveness and, therefore, the availability of support. Another reason is their effectiveness in accurately modelling plants – which, in conjunction with their pure simplicity, makes them a viable option for the less experienced graphics practitioner.

Procedural modelling was chosen because, again, it is a widely documented technique. It also allows for objects to be rendered only if they can be seen by the camera, an attribute which makes the creation of complex scenes less computationally intensive.

Both of these techniques are well-documented and widely used in the modelling and creation of geometrically complex scenes. This literature review is intended to be a representative study of the wide range of literature available on both topics. For completeness' sake, other plant modelling techniques are explored as well as instances of the use of procedural models in practice to model complex scenes. These scenes include forests, landscapes and cityscapes. A brief look at the software products available for L-System creation and rendering completes this review.

References

1. AMBURN, P., GRANT, E. and WHITTED, T. 1986. Managing Geometric Complexity with Enhanced Procedural Models. *SIGGRAPH 1986 Conference Proceedings,* pp. 189-195

2. BAKER, R. and HERMAN, G. 1970. CELIA – a cellular linear iterative array simulator. *Proceedings of the Fourth Conference on Applications of Simulation*, pp. 64—73

3. BIRCH, P., BROWNE, S., JENNINGS, V., DAY, A. and ARNOLD, D. 2003. Rapid Procedural Modelling of Architectural Structures. *SIGGRAPH 2003 Conference Proceedings*

4. BORCHERT, R. and HONDA, H. 1984. Control of development in the bifurcating branch system of *Tabebuia rose*a: A computer simulation. *Botanical Gazett*e 145(2), pp. 184–195

5. BORCHERT, R. and SLADE, N. 1981. Bifurcation ratios and the adaptive geometry of trees. *Botanical Gazette* 142(3), pp. 394–401

6. DE REFFYE, P., EDELIN, C., FRANÇON, J., JAEGER, M. and PUECH, C. 1988. Plant Models Faithful to Botanical Structure and Development. *Computer Graphics* 22(4), pp. 151-158

7. DEUSSEN, O., HANRAHAN, P., LINTERMANN, B., MĚCH, R., PHARR, M. and PRUSINKIEWICZ, P. 2002. Realistic Modeling and Rendering of Plant Ecosystems. *SIGGRAPH 2002 Conference Proceedings*

8. FOLEY, D. 1996. *Computer Graphics Principles and Practice Second Edition.* Addison-Wesley: Reading, Massachusetts

9. GREUTER, S., PARKER, J., STEWART, N. and LEACH, G. 2003. Undiscovered Worlds. *melbourneDAC - 5th International Digital Arts and Culture Conference.* Available online at http://hypertext.rmit.edu.au/dac/papers/Greuter.pdf

10. GRUBERT, M. 1999. *L-Arbor.* Master's Thesis, Technical University of Berlin, Germany. Available online at http://www.geocities.com/grubertm/id34.htm

11. GUERRAZ, S., PERBET, F., RAULO, D., FAURE, F. and CANI, M. 2003. A Procedural Approach to Animate Interactive Natural Sceneries. *CASA03.* Available online at http://www-imagis.imag.fr/Publications/2003/GPRFC03

12. HALLÉ, F., OLDEMAN, R. and TOMLINSON, P. 1978. *Tropical trees and forests.* Springer-Verlag: Heidelberg, New York.

13. HAMMEL, M. and PRUSINKIEWICZ, P. 1996. Visualization of developmental processes by extrusion in space time. *Proceeding of Graphics Interface '96*, pp. 246-258. Available online at http://www.cpsc.ucalgary.ca/Research/bmv/papers/

14. HANAN, J. 1992. *Parametric L-systems and their application to the modelling and visualization of plant*s. PhD thesis, University of Regina, Canada

15. HERMAN, G. and ROZENBERG, G. 1975 *Developmental systems and languages.* North-Holland: Amsterdam

16. HONDA, H., TOMLINSON, P. and FISHER, J. 1981. Computer simulation of branch interaction and regulation by unequal flow

rates in botanical trees. *American Journal of Botany* 68, pp. 569–585

17. JAMES, M., HANAN, J. and PRUSINKIEWICZ, P. 1993. *CPFG version 2.0 User's Manual.* Manuscript. Department of Computer Science, University of Calgary, Canada

18. JANSSEN, J. and LINDENMAYER, A. 1987. Models for the Control of Branch Positions and Flowering Sequences of Capitula in *Mycelis muralis* (L.) Dumont (Compositae). *New Phytologist* 105(2), pp. 191-220

19. JESCHKE, S., BIRKHOLZ, H. and SCHUMANN, H. 2003. A Procedural Model for Interactive Animation of Breaking Ocean Waves. *WSCG 2003*. Available online at http://wscg.zcu.cz/wscg2003/Papers_2003/K11.pdf

20. KURTH, W. 1994. *GROGRA*. Institute of Forest Biometry and Informatics. Faculty of Forest Sciences and Forest Ecology at the University of Göttingen. Available online at http://www.uni-forst.gwdg.de/~wkurth/grogra.html

21. LAPRÉ, L. 1993. Lparser. Available online at http://home.wanadoo.nl/laurens.lapre/

22. LINDENMAYER, A. 1968. Mathematical Models for Cellular Interaction in Development. *Journal of Theoretical Biology*

23. LÜCK, J., LÜCK, H. and BAKKALI, M. 1990. A comprehensive model for acrotonic, mesotonic, and basitonic branching in plants. *Acta Biotheoretica* 38, pp.257–288

24. MACKRILL, B. 2003. *Real-Time Rendering of Realistic Landscape*. Undergraduate project dissertation.

25. MACRI, D. and PALLISTER, K. 2004. *Procedural 3D Content Generation*. Technical report. Intel Developer Service. Available online at http://www.devx.com/Intel/Article/20182

26. MCCORMACK, J. 1993. Interactive Evolution of L-System Grammars for Computer Graphics Modelling. In: GREEN, D. and BOSSOMAIER, T. (Eds.): *Complex Systems: from Biology to Computation*, ISO Press: Amsterdam, pp. 118-130. Available online at http://www.csse.monash.edu.au/~jonmc/resources/L-systems-evol.pdf

27. MEYEROWITZ, E. 1994. The genetics of flower development. *Scientific American* 271(5), pp. 56–65

28. MOCK, K. 1998. Wildwood*. International Conference on Evolutionary Computing (ICEC '98).* Available online at http://www.math.uaa.alaska.edu/~afkjm/papers/Wildwood.doc

29. O'DONNELL, T. and OLSON, A. 1981. GRAMPS – A Graphics Language Interpreter for Real-Time, Interactive, Three-Dimensional Picture Editing and Animation. *Computer Graphics* 15(3), pp.133-142

30. OPPENHEIMER, P. 1986. Real-Time Design and Animation of Fractal Plants and Trees. *Computer Graphics* 20(4), pp. 55-64

31. PARISH, Y. and MÜLLER, P. 2001. Procedural Modeling of Cities. *SIGGRAPH 2001 Conference Proceedings*, pp. 301-308

32. PRUSINKIEWICZ, P. 1986. Graphical applications of L-systems. *Proceedings of Graphics Interface '86 — Vision Interface '86*, pp. 247–253

33. PRUSINKIEWICZ, P. 1993. *CPFG 2.0.* Virtual Laboratory, University of Calgary, Canada

34. PRUSINKIEWICZ, P. 1998. *L-Studio*. University of Calgary Computer Science, Canada. Available online at http://www.cpsc.ucalgary.ca/Research/bmv/l studio/

35. PRUSINKIEWICZ, P. and LANE, B. 2002. Generating spatial distributions for multilevel models of plant communities. *Proceedings of Graphics Interface,* pp. 69-80. Available online at http://www.cpsc.ucalgary.ca/Research/bmv/ papers/

36. PRUSINKIEWICZ, P. and LINDENMAYER, A. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag: New York

37. PRUSINKIEWICZ, P. and MĚCH, R. 1996. Visual Models of Plants Interacting with Their Environment. *SIGGRAPH 1996 Conference Proceedings*, pp. 397-410

38. PRUSINKIEWICZ, P., HAMMEL, M. and MĚCH, R. 1995. The Artificial Life of Plants. *Artificial Life for Graphics, Animation, and Virtual Reality*, v7 of SIGGRAPH 1995 Course Notes, pp 1-1 – 1-38, ACM Press 95. Available online at http://www.cpsc.ucalgary.ca/Research/bmv/ papers/l-sys.sig95.html

39. PRUSINKIEWICZ, P., HAMMEL, M. and MJOLSNESS, E. 1993. Animation of Plant Development. *SIGGRAPH 1993 Conference Proceedings*.

40. PRUSINKIEWICZ, P., HAMMEL, M., HANAN, J. and MĚCH, R. 1996. Visual models of plant development. In: ROZENBERG, G. and SALOMAA, A. (Eds.): *Handbook of formal languages*. Springer-Verlag: Berlin, pp. 535-597

41. PRUSINKIEWICZ, P., HANAN, J. and MĚCH, R. 1999. An L-system-based plant modeling language. In: NAGL, M., SCHUERR, A. and MUENCH, M. (Eds): *Applications of graph transformations with industrial relevance*. Springer: Berlin, pp. 395-410

42. PRUSINKIEWICZ, P., JAMES, M. and MĚCH, R. 1994. Synthetic Topiary. *SIGGRAPH 1994 Conference Proceedings*

43. PRUSINKIEWICZ, P., JIRASEK, C. and MOULIA, B. 2000. Integrating biomechanics into developmental plant models expressed using L-systems. In: SPATZ, H. and SPECK, T. (Eds.): *Plant biomechanics 2000.* Georg Thieme Verlag: Stuttgart, pp. 615-624 Available online at http://www.cpsc.ucalgary.ca/Research/bmv/ papers/

44. PRUSINKIEWICZ, P., LANE, B., MUENDERMANN, L. and KARWOWSKI, R. 2001. The use of positional information in the modeling of plants. *SIGGRAPH 2001 Conference Proceedings*, pp. 289-300

45. PRUSINKIEWICZ, P., LINDENMAYER, A. and HANAN, J. 1988. Developmental Models of Herbaceous Plants. *Computer Graphics* 22(4), pp. 141-150

46. RODKAEW, Y., LURSINSAP, C., FUJIMOTO, T., SURIPANT, S., CHONGSTITVATANA, P. and CHIBA, N. 2002. Modelling Leaf Shapes Using L-Systems and Genetic Algorithms. *International Conference NICOGRAPH*. April, Japan. Available online at http://www.cp.eng.chula.ac.th/faculty/pjw/pa per/nico2002.pdf

47. SMITH, A. 1984. Plants, fractals and formal languages. *Computer Graphics* 18(3), pp. 1-10

48. SORRENSEN-COTHERN, K., FORD, E. and SPRUGEL, D. 1993. A model of competition incorporating plasticity through modular foliage and crown development. *Ecological Monographs* 63(3), pp. 277-304

49. SZILARD, A. L. and QUINTON, R. E. 1979. An interpretation for D0L systems by computer graphics. *The Science Terrapin* 4, pp. 8–13

50. WEBER, J. and PENN, J. 1995. Creation and rendering of realistic trees. *SIGGRAPH 1995 Conference Proceedings*, pp. 119-128

51. WONKA, P., WIMMER, M., SILLION, F. and RIBARSKY, W. 2003. Instant Architecture. *SIGGRAPH 2003 Conference Proceedings,* pp. 669-677