Procedural Modelling of Plant Scenes

Short Paper Draft

Computer Science Honours Project

Author: Kim Randleff-Rasmussen (00R1008)

Supervisors: Shaun Bangay, Adele Lobb

1 Introduction

This paper describes a procedural model for the creation of complex plant scenes such as forests and jungles. The aim is to use the wide-spread technique of L-Systems to model the plants as this will mean that any aspect of the plant growth can be rendered without intensive modelling work on the part of the animator.

This project aims to model realistic complex organic environments by modelling not only the inhabitants of these environments but also the interactions between same. These interactions are modelled using a domination algorithm which stunts the growth of plants which are encroaching on the space of others. This same algorithm allows for the dynamic addition and removal of individuals to and from the population.

Previous work in the fields of plant modelling in general, L-Systems in particular and procedural modelling is looked at in Section 2. The specific approaches chosen, as well as reasons for these choices, are explained in Section 3. Section 4 looks at results that have been achieved thus far, and Section 5 looks at possible extensions to the final project.

2 Previous Work

2.1 Plant Modelling

Many techniques have developed over the years to model plant structure and growth. A well-known example is the mathematical model developed by de Reffye *et al.* (DE REFFYE et al. 1988) which formalises known botanical rules. Other models have focused on specific geometric phenomena such as (BORCHERT and HONDA 1984) which models the tendency of a tree to increase the vigour of growth after the loss of a branch.

(MEYEROWITZ 1994), (JANSSEN and LINDENMAYER 1987) and (BORCHERT and SLADE 1981) use specific species to create their models. A comprehensive study of various species can be found in (HALLE et al. 1978). This book asserts that the thousands of plant species in the world can be modelled using a mere twenty-three models.

Environmental influences obviously have a tremendous effect on plant growth and shape. (PRUSINKIEWICZ et al. 2000) looks at the effect of gravity on the branching behaviour of certain trees while (PRUSINKIEWICZ et al. 1994) looks at pruning - forced and self-imposed - and how this affects branch growth. (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. (HONDA et al. 1981) focuses on how interaction between individuals affects branching and (SORRENSEN-COTHERN et al. 1993) looks at interaction between modules of a single individual. (WEBER and PENN 1995) introduces a number of rules which ease the creation of models of various species.

2.2 L-Systems

L-Systems (Lindenmayer Systems, named for Hungarian biologist Aristid Lindenmayer) were developed in 1968 as a method for representing biological growth (LINDENMAYER 1968). They are replacement languages which use defined production rules to supplant symbols within a string. Each symbol in the strings represents an addition or change to the graphical representation of the organism the string describes.

The most common interpretation of L-System symbols (or modules) is through Turtle graphics. In this case, a basic L-System alphabet consists primarily of:

F(s) - Move forward a step of length *s* and draw a line segment from the original to the new position of the turtle.

+(θ) - Turn left by angle θaround the z axis.

-(θ) - Turn right by angle θaround the z axis.

&(θ) - Turn left by angle θaround the y axis.

^(θ) - Turn right by angle θaround the y axis.

/(θ) - Turn left by angle θaround the x axis.

\(θ) - Turn right by angle θaround the x axis.

| - Turn 180 degrees around the z axis.

[- Push the current state of the turtle (position, orientation and drawing attributes) onto a pushdown stack.

] - Pop a state from the stack and make it the current state of the turtle. No line is drawn although in general the position and orientation of the turtle are changed.

List taken from (PRUSINKIEWICZ et al. 1995)

Other letters and symbols may also be used to delineate areas for string replacement.

Another important element of L-Systems is the production rules which specify what symbols are to be replaced by which strings at each iteration step. Due to this recursive process, L-Systems are a succinct language for describing the growth of organic material such as plant. The production rules have the following structure:

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

where *lc* is the left-hand context, *rc* the right-hand context, *pred* the predecessor (the symbol to be replaced), *cond* a condition and *succ* the successor (the string to replace the predecessor). The rule is only run (i.e. the symbol replaced by the successor) if the condition evaluates to *true*. *prob* is a number which defines the probability of the rule being chosen. This is only found in stochastic L-Systems where more than one rule can exist for each symbol. Only the predecessor and the successor are compulsory.

For example, an L-System may comprise an axiom *F* and a replacement rule *F* → *F[<F][>F]F*. A first pass of the L-System will yield a new string *F[<F][>F]F* where the original *F* has been replaced. A second pass will yield *FF[<F][>F]F[<FF[<F][>F]F][>FF[<F][>F]F]F[<F][>F]F* where each *F* in the second string is replaced by the successor of the rule.

Many extensions to basic L-Systems have also been developed to add greater functionality to the basic technique. Table L-Systems were described by Rozenberg (ROZENBERG 1973) to allow different production rules to be chosen according to certain environmental factors. Prusinkiewicz *et al.* (PRUSINKIEWICZ et al. 1994) developed another type of L-System - environmentally sensitive L-Systems - which uses query symbols (*?E(parameters)*) to provide environmental information to the plant models. In (PRUSINKIEWICZ and MECH 1996), bilateral communication was added which allowed the plant to return information to the environment.

Other extensions allowed for more than one individual plant to be modelled from the same L-System. The introduction of parametric L-Systems (PRUSINKIEWICZ et al. 1995) allowed certain aspects of the model to be changed dynamically, such as angle size and segment length. Differential L-Systems (or dL-Systems) (PRUSINKIEWICZ et al. 1993) extend these parametric L-Systems by incorporating continuous time flow which allows for smooth animation of plant development.

Stochastic L-Systems (EICHHORST and SAVITCH 1980) use multiple productions for each symbol, the specific rule being chosen according to assigned probabilities. This functionality allows a single L-System to yield a number individual plants with the same basic shape but with noticeable differences. Erasing productions (HERMAN and ROZENBERG 1975) also allow for individual variation within a single model.

The addition of genetic algorithms (MCCORMACK 1993), mutation (MOCK 1998) and evolution (RODKAEW et al. 2002) into L-Systems has allowed for realistic modelling of plant development over long periods of time. Extrusion in space time (HAMMEL and PRUSINKIEWICZ 1996) extends a two-dimensional structure's development into a three-dimensional line or curve which represents the progress over time.

L-Systems have also been used to model entire plant populations. An example of this use can be found in (PRUSINKIEWICZ and LANE 2002) which introduces multi-set L-Systems which allow for production rules to be applied to a set of strings instead of to a single string.

2.3 Procedural Modelling

Procedural modelling is a computer graphics technique designed to decrease the computational power required to render a geometrically complex scene. It does so by improving on two areas of scene rendering. Firstly, objects that are not within the view spectrum of the camera are not rendered (FOLEY 1996). Secondly, elements of the model are rendered 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.

(AMBURN et al. 1986) suggests 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.

Procedural modelling has long been in use in programs which model geometrically complex scenes such as forests, cities and crowds. Large cities, for example, have been procedurally modelled in (BIRCH et al. 2003), (GREUTER et al. 2003), (PARISH and MULLER 2001) and (WONKA et al. 2003). In terms of forests or landscape generation, some prominent projects are described in (GUERRAZ et al. 2003) and (DEUSSEN et al. 1998).

3 Design

3.1 Procedural Models

The basis of procedural modelling is to have all of the complex geometry needed in a scene generated within a procedure. In this procedural model, the entire scene is modelled by a single procedure (*forest*). Within this procedure, only the placement and size of the plants is calculated. The actual geometry generation is delegated to another procedure (*lsystem*) which, in turn, delegates the generation to another (*segment*). This technique means that large scenes can be developed quickly and efficiently as the geometry generation is only tackled after the other calculations.

Each procedure is supplied with the Transformation matrix from the calling procedure. Along with this, the procedure is also supplied with a long integer random key which can be used to seed a pseudo-random number sequence. It is useful to use the same number to seed the sequence so that the exact same forest can be generated provided the key is known.

3.2 L-Systems

An L-System is read from a *.sys* file which has the following structure:

LSYSTEM file header

r an integer denoting the number of replacement rules

A axiom (or L-System start string)

R replacement rules with the structure *A R p* (*A* is the symbol to replace, *R* the string to replace it with and *p* is a double value which signifies the probability of the rule being fired).

The L-System strings within this file make use of the following alphabet:

F(l,w,t) - Draw a tapered rectangular segment of length *l*, base width *w* and tapered tip width *t*.

<(θ) - Turn right by angle θ around the z axis.

>(θ) - Turn left by angle θ around the z axis.

&(θ) - Turn right by angle θ around the x axis.

@(θ) - Turn left by angle θ around the x axis.

%(θ) - Turn right by angle θ around the y axis.

#(θ) - Turn left by angle θ around the y axis.

[...] - Branch by pushing (and later popping) the current Transformation matrix to (from) a pushdown stack.

L-System strings are stored in a defined LSystem data structure which allows for easy access to any of the modules within the string and their parameters.

3.3 Plant Placement

(PRUSINKIEWICZ and LANE 2002) introduces a number of methods by which the placement of plants in a forest can be determined. These methods can be considered to be local-to-global or global-to-local. The latter category contains methods which place individuals according to pre-defined and constantly updated density functions. As each plant is placed, the probability function is updated according to this new placement and the placement of the next plant will be affected.

Local-to-global plant placement methods simulate ecosystem interaction by focusing on each individual. Multi-set L-Systems are an example of this type of placement methodology. These L-Systems apply a set of production rules to a set of L-Systems strings (which symbolises plant population). This allows for the dynamic addition and removal of individual plants and the dynamic upkeep of the population as a whole.

(DEUSSEN et al. 1998) mentions two methods for placing plants within a modelled forest. The first of these involves inputting a density map to which an error diffusion algorithm is then applied. The specific algorithm implemented in this paper is one introduced by Floyd and Steinberg. The second method places plants randomly and then iteratively grows or kills individuals based on domination. It is this second method which is used in this project.

3.4 Generalised Subdivision Meshes

(MAIERHOFER 2002) introduces generalised subdivision meshes. This technique allows for the smoothing of meshes using a combination of a number of common subdivision methods (such as Catmull-Clark, Doo-Sabin and $\sqrt{3}$-Subdivision). It also uses procedural modelling to model small scale surface detail. The focal point of the thesis is rule-based mesh growing which applies L-Systems to mesh faces as opposed to symbols.

4 Implementation

In th creation of a forest, the plant placements are randomly calculated. These random placements are seeded by the long integer key which is passed to the procedural model (see Section 3.1). This key then

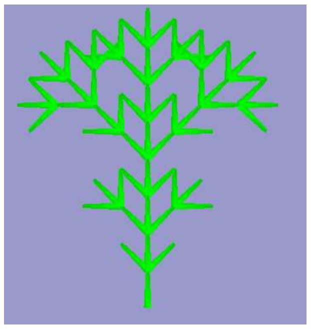

Figure 1: An example of an L-System tree

seeds the *x* and *z* coordinates of the plant to be placed. The same key seeds a random number which determines the L-System to be used in the determined place and, hence, the plant to be placed in that position. Figure 2 shows an example of the random placement of plants that can be generated.

A final random number determines the starting iteration for the plant to be modelled. Later, this number will be increased according to the frame being rendered.

Once the plant placement has been determined, an LSystem must be created and sent to the *lsystem* procedure for parsing and translation. The creation occurs when a *.sys* file is read and the full L-System string generated according to the number of iterations desired. The *lsystem* procedure then parses this string, changing the Transformation matrix according to the >, <, @, &, %, # and | modules and calling the *segment* procedure in response to an *F* module. Figure 1 shows an example of an L-System image where the originating string is:

F[>F][<F]F[>F[>F][<F]F][<F[>F][<F]F]F[>F][<F]F[>F[>F][<F]F[>F[>F][<F]F][<F[>F][<F]F]F[>F][<F]F][<F[>F][<F]F[>F[>F][<F]F][<F[>F][<F]F]F[>F][<F]F]F[>F][<F]F[>F[>F][<F]F][<F[>F][<F]F]F[>F][<F]F.

Generalised mesh techniques are implemented in this project in order to create more realistic, smooth trees. In order to achieve this, a *segment* is repeatedly smoothed from its original square shape. Figure 3 shows a section of an L-System tree before application of these techniques and after smoothing.

5 Extensions

In (TOBLER et al. 2002), a method is explained by which texture coordinates from a generated complex mesh are used to displace the vertices. This method allows for grooves and ridges to be added to such complex meshes as trees, thus increasing the realism of the final image. This is a possible extension to the project described above.

Figure 2: A section of a forest showing the random tree placement

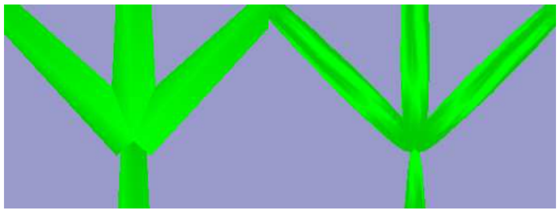

Figure 3: Two versions of the same joint showing the difference between the generated cube (left) and the same cube after smoothing (right)

Another possible extension is the definition of the L-System files using a grammar definition language such as COCO/R. This would be relatively easy to implement and would streamline the process of parameter finding and symbol replacement.

References

AMBURN, P., E. GRANT and T. WHITTED (1986). Managing geometric complexity with enhanced procedural models. In *SIGGRAPH 1986 Conference Proceedings* (1986), ACM Press, pp. 189–195.

BIRCH, P., S. BROWNE, V. JENNINGS, A. DAY and D. ARNOLD (2003). Rapid procedural modelling of architectural structures. In *SIGGRAPH 2003 Conference Proceedings* (2003), ACM Press.

BORCHERT, R. and H. HONDA (1984). Control of development in the bifurcating branch system of tabebuia rosea: A computer simulation. *Botanical Gazette 145*, 2 (1984), pp. 184–195.

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

DEUSSEN, O., P. HANRAHAN, B. LINTERMANN, R. MECH, M. M. PHARR and P. PRUSINKIEWICZ (1998). Realistic modeling and rendering of plant ecosystems. In *SIGGRAPH 1998 Conference Proceedings* (1998), ACM Press.

EICHHORST, W. and J. SAVITCH (1980). Growth functions of stochastic lindenmayer systems. *Information and Control 45,* 3 (1980), pp. 217–228.

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

GREUTER, S., J. PARKER, N. STEWART and G. LEACH (2003). Undiscovered worlds. In *melbourneDAC - 5th International Digital Arts and Culture Conference* (2003).

GUERRAZ, S., F. PERBET, D. RAULO, F. FAURE and M. CANI (2003). A procedural approach to animate interactive natural sceneries. In *CASA03* (2003).

HALLE, F., R. OLDEMAN and P. TOMLINSON (1978). *Tropical trees and forests*. Springer-Verlag, Heidelberg, New York, 1978.

HAMMEL, M. and P. PRUSINKIEWICZ (1996). Visualization of developmental processes by extrusion in space time. In *Proceeding of Graphics Interface '96* (1996), pp. 246–258.

HERMAN, G. and G. ROZENBERG (1975). *Developmental systems and languages*. North-Holland, Amsterdam, 1975.

HONDA, H., P. TOMLINSON and J. FISHER (1981). Computer simulation of branch interaction and regulation by unequal flow rates in botanical trees. *American Journal of Botany*, 68 (1981), pp. 569–585.

JANSSEN, J. and A. LINDENMAYER (1987). Models for the control of branch positions and flowering sequences of capitula in mycelis muralis (l.) dumont (compositae). *New Phytologist 105,* 2 (1987), pp. 191–220.

LINDENMAYER, A. (1968). Mathematical models for cellular interaction in development. *Journal of Theoretical Biology* (1968).

MAIERHOFER, S. (2002). *Rule-Based Mesh Growing and Generalised Subdivision Meshes*. PhD thesis, Vienna University of Technology, Austria, 2002.

MCCORMACK, J. (1993). Interactive evolution of l-system grammars for computer graphics modelling. In *Complex Systems: from Biology to Computation*, D. Green and T. Bossomaier, Eds. ISO Press, Amsterdam, 1993.

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

MOCK, K. (1998). Wildwood. In *International Conference on Evolutionary Computing (ICEC '98)* (1998).

PARISH, Y. and P. MULLER (2001). Procedural modeling of cities. In *SIGGRAPH 2001 Conference Proceedings* (2001), ACM Press, pp. 301–308.

PRUSINKIEWICZ, P., C. JIRASEK and B. MOULIA (2000). Integrating biomechanics into developmental plant models expressed using l-systems. In *lant biomechanics 2000*, H. Spatz and T. Speck, Eds. Georg Thieme Verlag, Stuttgart, 2000.

PRUSINKIEWICZ, P. and B. LANE (2002). Generating spatial distributions for multilevel models of plant communities. In *Proceedings of Graphics Interface* (2002), pp. 69–80.

PRUSINKIEWICZ, P., B. LANE, L. MUENDERMANN and R. KARWOWSKI (2001). The use of positional information in the modeling of plants. In *SIGGRAPH 2001 Conference Proceedings* (2001), ACM Press, pp. 289–300.

PRUSINKIEWICZ, P. and R. MECH (1996). Visual models of plants interacting with their environment. In *SIGGRAPH 1996 Conference Proceedings* (1996), ACM Press, pp. 397–410.

PRUSINKIEWICZ, P., R. MECH and M. HAMMEL (1995). The artificial life of plants. In *Artificial Life for Graphics, Animation, and Virtual Reality, v7 of SIGGRAPH 1995 Course Notes* (1995), ACM Press, pp. 1.1–1.38.

PRUSINKIEWICZ, P., R. MECH and M. JAMES (1994). Synthetic topiary. In *SIGGRAPH 1994 Conference Proceedings* (1994), ACM Press.

PRUSINKIEWICZ, P., E. MJOLSNESS and M. HAMMEL (1993). Animation of plant development. In *SIGGRAPH 1993 Conference Proceedings* (1993), ACM Press.

REFFYE, P. DE, C. EDELIN, J. FRANCON, M. JAEGER and C. PEUCH (1988). Plant models faithful to botanical structure and development. *Computer Graphics 22*, 4 (1988), pp. 151–158.

RODKAEW, Y., C. LURSINSAP, T. FUJIMOTO, S. SURIPANT, P. CHONGSTITVATANA and N. CHIBA (2002). Modelling leaf shapes using l-systems and genetic algorithms. In *International Conference NICOGRAPH* (April, Japan, 2002), April, Japan.

ROZENBERG, G. (1973). T0l-systems and languages. *Information and Control 23*, 4 (1973), pp. 357–381.

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

TOBLER, R., S. MAIERHOFER and A. WILKIE (2002). A multiresolution mesh generation approach to procedural definition of complex geometry. In *International Conference on Shape Modeling and Applications 2002 (SMI'02* (Banff, Canada, 2002), Banff, Canada, pp. 35–42.

WEBER, J. and J. PENN (1995). Creation and rendering of realistic trees. In *SIGGRAPH 1995 Conference Proceedings* (1995), ACM Press, pp. 119–128.

WONKA, P., M. WIMMER, F. SILLION and W. RIBARSKY (2003). Instant architecture. In *SIGGRAPH 2003 Conference Proceedings* (2003), ACM Press, pp. 669–677.