A Global Signal Propagation Technique for the Modeling of Plants

Pedro Luiz Eira Velha, Antonio Elias Fabris
Computer Graphics and Applied Computational Geometry Project - CGCAP
Instituto de Matemática e Estatística, Universidade de São Paulo
Caixa Postal 66281, 05315-970, São Paulo - SP, Brazil
{eira, aef}@ime.usp.br

ABSTRACT

This paper introduces a new technique for modeling signal and nutrient propagation in plant models to counteract the main disadvantages of current methods: the difficulty of setting the velocity of the propagation and the complexity of models with many signals. This technique uses implicit parameters and global propagation functions, offering a better control over the velocity of propagation, making the modeling of many independent signals an easy task, and keeping the model simple and objective. Furthermore, our approach is easy to be aggregated into existing models and makes the modeling of signal and nutrient propagation a more intuitive task.

Keywords

Signal propagation, nutrient propagation, L-system, plant modeling, simulation

1 INTRODUCTION

The modeling and simulation of plant development play an important role in computer graphics and in biology. Beyond the rendering of realtime [Die97a, Tra97a] and realistic [Deu98a, Ger96a] nature scenes, plant modeling is useful for helping scientists to have both a better understanding of the plant development [Pru93b] and the interaction between its organs [Pru96a]. It is also an excellent tool for testing mathematical models of biology, like the effect of pruning [Pru94c] or the reaction of plants to changes in the water concentration in the soil [Mec96b].

Current techniques made it possible to describe the development of plants by defining the rules of its internal processes, like the water, nutrient and photosynthesis flow, as long as its interaction with the external environment. Our approach to the modeling of plants is the well known Lindenmayer Systems (L-Systems) which provides a procedural method to model almost any kind of branching structures [Fow92a, Par01a].

In section 2 we briefly describe the L-System structure and, in section 3, we present current methods used to simulate signal propagation in plant models. In section 4, we introduce a new technique for modeling signal propagation which offers a better control over the velocity of the propagation, makes the modeling of many independent signals an easy task, and keeps the model simple and objective. The presented technique is also easy to be aggregated into existing models and makes the modeling of signal and nutrient propagation a more intuitive task. In section 5, we present three examples to illustrate the benefits brought by this new technique and in section 6 we briefly describe the modeling environment used. Finally, in section 7, we comment the results obtained and show some topics that are open for further research.

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

2 LINDENMAYER SYSTEMS

In this section, we summarize the main features of L-Systems pertinent to this paper. A more extensive exposition of L-Systems can be found in [Pru96a, Pru90d].

We call *modules* the most basic structures of a plant. The modules usually represent repeating structures of the plant, like flowers, leaves and branch segments. The modules may be followed by parameters that quantify some characteristics of the modules, like its size, age or the concentration of certain substances. The modules are written as a letter (or a word) followed by the parameters inside parentheses. A plant is then described as a sequence of modules with each ramification delimited by a matched pair of brackets. This representation is called *bracketed string*.

The visualization of the structure is done using *turtle interpretation* [Pru96a]: the bracketed string is scanned from left to right and some selected modules are considered as commands that maneuver a LOGO-style turtle. The main selected modules used in this paper are listed below:

$F(s)$	draws a line segment of length s.
$!(w)$	sets the current line width to w.
$+(\alpha), -(\alpha)$	rotates $\pm\alpha$ around the up vector $\vec{U}$.
$/(\alpha), \backslash(\alpha)$	rotates $\pm\alpha$ around the heading vector $\vec{H}$.
$\&(\alpha), \wedge(\alpha)$	rotates $\pm\alpha$ around the left vector $\vec{L}$.
$\sim S$	draws the predefined bezier surface S.
$@O(d)$	draws a sphere of diameter d.

If the parameters of the reserved modules are omitted, the modules use a default value. This is the most common notation used to describe L-Systems and was introduced by Prusinkiewicz [Pru90d]. Figure 1 shows an example of the turtle interpretation of the following bracketed string:

$$F(1)[-(45) \sim leaf]F(1)[+(45) \sim leaf]F(2) \sim flower$$

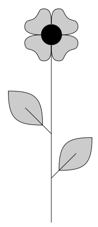


Figure 1: a flower.

An L-System describes the development of a plant model with a set of rewriting rules called *productions*, that replaces each module with a set of successor modules representing the next stage in the development of that specific part of the plant. The L-Systems differ from other rewriting systems by the fact that the rules are applied in parallel to all modules of the plant. This is motivated by the biological fact that the cellular division occurs simultaneously in many cells.

An L-System is composed by one *axiom* that defines the initial state of the plant and a set of *productions*. The productions are described using the following syntax:

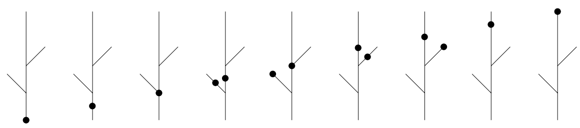
$$lc < pred > rc : cond \rightarrow succ$$

A production replaces the predecessor module $pred$ with zero, one or more successor modules $succ$, as long as the predecessor is preceded by the left context lc and succeeded by the right context rc, and the condition $cond$ evaluates to $true$. Only the predecessor and the successor are mandatory fields. Actually, L-Systems with no left or right contexts are called *context free*, while L-Systems with one or two contexts are called *context sensitive* L-Systems. The process of substitution of all modules of a plant model by its successors is called *derivation*.

3 SIGNAL PROPAGATION IN PLANT MODELS

One important aspect of plant development is the communication between the plant organs. The signal propagation is among the most important features that increase the expressive power of the L-Systems. It represents the information flow through the plant, like the flow of the nutrients extracted from the soil and carried up towards the branches, or the photosynthesis flow from the leaves down towards the base of the tree. The signal propagation is called *acropetal* when the signal is carried up towards the top of the tree, and is called *basipetal* when the signal is carried down towards the base of the tree.

As described in [Pru96a, Pru90d], there are two ways to represent a signal in a plant model: with a module, and with a parameter. The first approach is used when we want to model a discrete signal, like the presence of a hormone that triggers the transformation of a bud to a flower. The following L-System shows an example of an acropetal signal propagation where the signal is represented by the module $@O$:



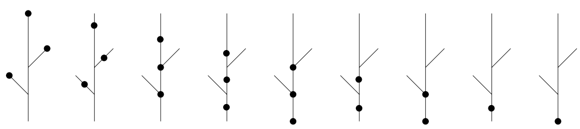
$$w : \quad @O\ FF[+FF]FF[-FF]FFFF$$

$$p_1 : \quad @O < F \quad \rightarrow \quad F\ @O$$
$$p_2 : \quad @O \quad \rightarrow \quad \epsilon$$

Figure 2: Acropetal signal propagation.

The initial state of the plant is set by the axiom w as a branching structure with the signal module $@O$ in its base. Production p_1 replaces and internode F preceded by a signal $@O$ with an internode followed by a signal. Production p_2 removes the signal modules from the previous derivation.

The following L-System shows a similar model simulating the propagation of a basipetal signal:



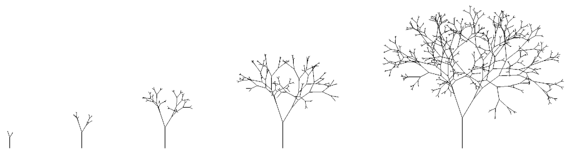
$$w : \quad FF[+FF\ @O]FF[-FF\ @O]FFFF\ @O$$

$$p_1 : \quad F > @O \quad \rightarrow \quad @O\ F$$
$$p_2 : \quad @O \quad \rightarrow \quad \epsilon$$

Figure 3: Basipetal signal propagation.

The axiom w sets the initial state of the plant as a branching structure with the signal module $@O$ in its apices. Production p_1 replaces and internode F followed by a signal $@O$ with an internode preceded by a signal. Production p_2 removes the signal modules from the previous derivation.

The second approach to simulate a signal in a plant model is to represent the signal as a parameter of the modules. This technique is used to model quantifiable values, like the concentration of nutrients in each module. The following L-System shows a tree model that grows according to the nutrient concentration in its branches.



$$
\begin{array}{ll}
w: & R\,F(1,0)\,A \\
p_1: & R < F(d,n) \quad \rightarrow \quad F(d+1,\ n+1) \\
p_2: & F(d_1,n_1) < F(d,n) : n_1 > n \\
 & \quad \rightarrow \quad F(d+1,\ \frac{n+n_1}{2}) \\
p_3: & F(d,n) < A : n \geq 5 \\
 & \quad \rightarrow \quad /(134)[+(38)F(1,0)A][-(27)F(1,0)A]
\end{array}
$$

Figure 4: Tree model simulating the acropetal nutrient flow from the root of the tree towards its branches. The figure shows the structure after 10, 15, 20, 25 and 30 derivations.

The axiom w sets the initial state of the plant as an internode F followed by an apex A and preceded by a module R representing the tree root. The second parameter of the module F represents the nutrient concentration in that module. Production p_1 describes the nutrient flow from the root towards the first internode of the tree, and production p_2 describes the nutrient flow between two internodes. When the nutrient concentration near an apex reaches the value 5, production p_3 bifurcates the apex into two new branches.

The techniques to model signal and nutrient propagation presented in this section have some disadvantages:

- It is difficult to set the velocity of the propagation: in the first technique, the signal propagates in discrete steps making it difficult, for example, to model a signal that traverses 7 modules in 11 derivations. Furthermore, it is difficult to model a signal that traverses more than one module in each derivation.
- It is hard to add a new signal into an existing model: this happens because a new parameter may have to be added to all occurrences of certain modules, and because some productions may need to propagate the signal while doing their previous tasks.
- L-Systems with many signals are hard to understand because the propagations may be described in the same productions where important features of the model are specified. Furthermore, its easy to get confused when working with modules with many parameters.

4 GLOBAL SIGNAL PROPAGATION TECHNIQUE

In this section we introduce a new technique for modeling signal and nutrient propagation in L-Systems. In this technique, the information flow is described by a global propagation function and the L-System describes only the reaction of the modules that are reached by the signal. Each signal is defined by four main elements:

- a *source module*, which is a special module that initiates the flow of the signal through the plant,
- an *implicit parameter*, that contains the current value of the signal in each module,
- a *propagation function*, that defines how the intensity of the signal varies as it flows through the plant, and
- a *direction*, that defines if the signal flow is acropetal (from the base towards the apices) or basipetal (from the apices down towards the root).

The signal propagation is given by the solution of the wave equation of the form $u(x,t) = F(x - vt)$, where x is the position of the signal source, t is time, and v is the velocity of the propagation [Fol95a]. The shape of the wave is given by the *propagation function*, which is the function $F(x)$ (at time $t = 0$). The position x of the signal source and the velocity v of the signal propagation are defined by the parameters of the *source module*: the first parameter is the value of the expression $x - vt$, and the second parameter is the value of the velocity v. In the initial state of the plant, the time t is equal to zero, so the first parameter $x - vt$ is equal to x. After each L-Systems derivation, the time t is incremented by one unit and the value of the first parameter of the source module is updated automatically. Figure 5 illustrates the propagation of a wave given by the function $F(x) = 2^{-x^2}$ and shows the values of the parameters of the source module in each derivation.

The distance from the signal source is measured along the branches of the plant (the first parameter of the $F(s)$ modules), and the value of the signal in each module is given by the *implicit parameter*. After each derivation, the value of the implicit parameter of all modules is automatically updated to the correct value of the propagation function for that module.

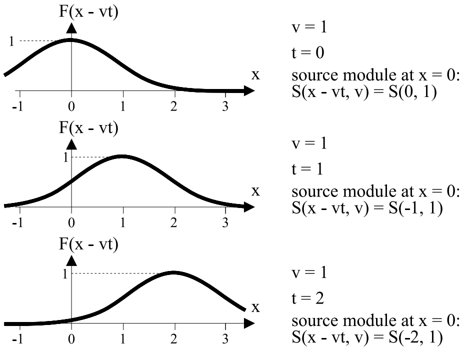


Figure 5: The propagation of the function $F(x - vt) = 2^{-x^2}$, with the source module S at position $x = 0$.

The implicit parameters are called *implicit* because they do not appear in every occurrence of the modules. They appear only in the productions where their values are used. All implicit parameters are denoted by the symbol $ followed by an identifier. The following example gives a typical production that uses the value of an implicit parameter:

$$A(v, c, s = \$s) : s == 1 \rightarrow \sim flower$$

In this example, the apex A has two regular parameters v and c and an implicit parameter $\$s$ which is loaded into variable s. When the implicit parameter $\$s$ reaches the value 1, this production replaces the apex A with a flower $\sim flower$.

It is possible to have more than one signal source module in the same plant. In this case, each source module propagates one function independently. The implicit parameter of each module is calculated by taking the sum of the values of all signals that reach the module. The Example 3 in the following section illustrates the use of more than one source module to simulate the photosynthesis flow from the leaves down toward the plant root. An L-System may also include more than one type of signal with its own source module and implicit parameter. In this case, the signals are completely independent.

5 EXAMPLES

In this section we present some examples of L-Systems with the introduced signal propagation technique to demonstrate the benefits of this new approach. The first example illustrates the use of different propagation functions in a plant model. The second example shows how the velocity of the propagation can be easily changed and how it affects the model. The third and last example illustrates the use of many different signals in the same model.

Example 1

This example shows how different propagation functions can change the appearance of a plant model. The following L-System describes a model of a tree with an acropetal signal representing the nutrient flow through the plant. The thickness of each segment of a branch will be given by the nutrient concentration in that segment.

signal: {
  source: S
  parameter: $\$s$
  function: $F(x)$
  direction: up
}

$$
\begin{array}{llll}
w: & S(0,1)A(10) & & \\
p_1: & A(x) : x > 0 & \rightarrow & !(1)FA(x-1) \\
p_2: & A(x) : x = 0 & \rightarrow & /(121)[+(38)!(1)FA(10)] \\
 & & & [-(17)!(1)FA(10)] \\
p_3: & !(x, s = \$s) & \rightarrow & !(s)
\end{array}
$$

The initial state of the plant is defined by the axiom w as an apex preceded by the signal source module S with propagation velocity set to 1. Productions p_1 and p_2 defines the growth of the apex: the apex grows linearly for ten derivations and then splits into two new branches. As the apex grows one unit in each derivation, the signal propagates at the same speed of the plant growth. Production p_3 adjusts the width of each internode of the tree by setting the first parameter of the set-width modules $!(x)$ to the value of the nutrient concentration ($\$s$) in that module. Figure 6 shows the development of this model for different propagation functions.

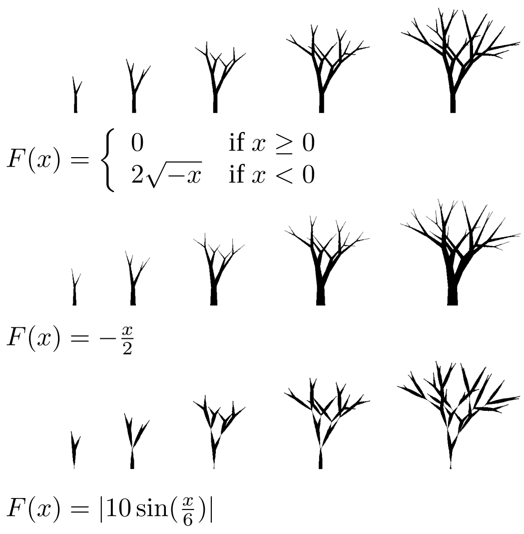


Figure 6: Tree models with the thickness of the branches specified by an acropetal signal given by function $F(x)$ and the structures after 20, 30, 40, 50 and 60 derivations.

Example 2

In this example we illustrate the effect of changes in the velocity of a signal propagation. The following L-System describes the model of a tree with an acropetal signal that defines the growth rate of the branches. The growth rate will be defined by the following propagation function:

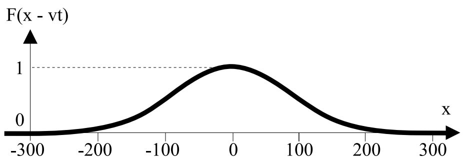


Figure 7: $F(x - vt) = 2^{-(\frac{x}{100})^2}$.

This function simulates a tree that grows slowly in its early ages, then accelerates in the middle of its development and level off when it comes close to its final size.

signal: {
  source: S
  parameter: $\$s$
  function: $2^{-(\frac{x}{100})^2}$
  direction: up
}

$$w: \quad S(300, v)F(1)A(9)$$

$$\begin{array}{llll} p_1: & A(x) : x > 0 & \rightarrow & A(x-1) \\ p_2: & A(x) : x = 0 & \rightarrow & /(121)[+(38)F(1)A(9)] \\ & & & [-(17)F(1)A(9)] \\ p_3: & F(d, s = \$s) & \rightarrow & F(d+s) \end{array}$$

The axiom w defines an internode F followed by an apex A and preceded by the signal source module S with initial x value set to -300 and propagation velocity set to the constant value v. The growth of the apex is defined by productions p_1 and p_2: after nine derivations the apex splits into two new branches. Production p_3 increases the length of each internode by the intensity of signal $\$s$. Figure 8 shows the development of the model for three different velocities.

In the first model of Figure 8 ($v = 3$), the branches of the tree develop early when the tree is still small and, when the tree grows, it keeps the same appearance as when it was young. In the second model ($v = 7$), the tree starts to grow with few branches and the crown of the tree is formed in the middle of its development. In the third model ($v = 10$) the tree reaches its full size with very few branches and then the tree crown is formed by many short branches that did not have time to grow.

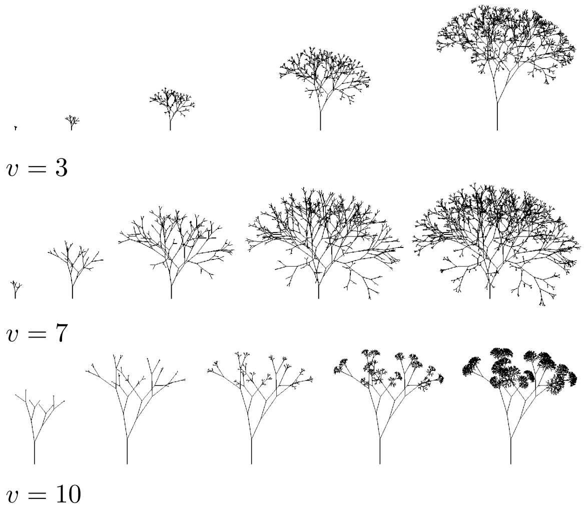


Figure 8: Tree models with the growth rate defined by an acropetal signal given by the propagation function $F(x - vt) = 2^{-(\frac{x}{100})^2}$ for different propagation velocities and the structures after 40, 60, 80, 100 and 120 derivations.

Example 3

This example illustrates the use of more than one signal in the same model. It describes a tree model with four different signals:

- The first one ($\$n$) is similar to the signal presented in the previous example: it is an acropetal signal that represents the nutrient flow through the plant and defines the growth rate of the branches. The model simulates a tree that grows slowly in its early ages, then accelerates in the middle of its development and then level off when it comes close to its final size.
- The second one ($\$p$) is a basipetal signal simulating the photosynthesis flow from the leaves of the tree down toward its root. This signal is used to set the thickness of the internodes, so that branches with more leaves are allowed to grow stronger than branches with fewer leaves.
- The third one ($\$s$) is an acropetal signal that informs the tree the current season. The tree reacts to season changes in the following way: in the fall, the leaves become orange and then fall; in the winter, the tree remains with no leaves; in the spring, the leaves reappear together with flowers; and, in the summer, the flowers disappear and the tree remains only with the leaves.
- The fourth one ($\$t$) is an acropetal signal that informs all modules the age of the tree. It is used to define when the flowers first appear and to define the size of the leaves.

signal: { /* Nutrient Concentration */
source: N
parameter: $\$n$
function: $(2^{-(\frac{x}{100})^2})/5$
direction: up
}

signal: { /* Photosynthesis */
source: $\sim leaf$
parameter: $\$p$
function: 1
direction: $down$
}

signal: { /* Season */
source: S
parameter: $\$s$
function: $(\frac{x}{100}) \bmod 20$
direction: up
}

signal: { /* Tree Age */
source: T
parameter: $\$t$
function: $\frac{x}{2000}$
direction: up
}

#define $fall(s)$ $(s < 5)$
#define $winter(s)$ $(s \geq 5$ and $s < 10)$
#define $spring(s)$ $(s \geq 10$ and $s < 15)$
#define $summer(s)$ $(s \geq 15)$
#define $leaves(s)$ $(s \geq 8)$

#define v 1.4 /* growth velocity */
#define a 30 /* apex bifurcation rate */
#define l_d 6 /* leaf duration */
#define f_d 4 /* flower duration */

w : $T(0, -100)S(0, -100)N(300, v)!(0)F(1)A(a)$

p_1 : $F(x, n = \$n) \rightarrow F(x + n)$

p_2 : $!(x, p = \$p) : x \leq \frac{\sqrt{p}}{4} \rightarrow !(x + 0.2)$

p_3 : $A(x) : x = 0 \rightarrow [+(38)!(0)F(1)A(a)][-(17)!(0)F(1)A(a-1)]$

p_4 : $A(x, s = \$s, t = \$t)$: $(t > 5)$ and $(x \bmod 2 = 0)$ and $spring(s)$ $\rightarrow /(123)[Fl(f_d)]A(x-1)$

p_5 : $A(x, s = \$s) : (x \bmod 2 = 1)$ and $leaves(s)$ $\rightarrow /(123)[Lf(l_d)]A(x-1)$

p_6 : $A(x) \rightarrow A(x-1)$

p_7 : $Fl(x) : x = f_d \rightarrow Fl(x-1)\ @v\ ,(2)\ \sim flower$

p_8 : $Fl(x) : x > 0 \rightarrow Fl(x-1)$

p_9 : $Fl(x) : x = 0 \rightarrow \%$

p_{10} : $Lf(x, t = \$t) : x = l_d \rightarrow Lf(x-1)\ @v\ @D(\sqrt{t})\ ,(1)\ \sim leaf(0, 0)$

p_{11} : $Lf(x) : x > 0 \rightarrow Lf(x-1)$

p_{12} : $Lf(x) : x = 0 \rightarrow \%$

p_{13} : $,(x, s = \$s) : x \leq 6$ and $fall(s) \rightarrow ,(s+3)$

The axiom w sets the initial state of the plant to an internode F of width zero followed by an apex A and preceded by the signal source modules of the acropetal signals (T, S, N). The growth of the internodes is defined by production p_1 and is given by the nutrient concentration $\$n$ in each internode. Production p_2 defines the thickness of each branch according to the photosynthesis produced by its leaves. Each leaf is a source module of the photosynthesis signal $\$p$ and propagates a constant function of value 1. In a branch with more than one leaf the signals are summed and the value of the photosynthesis in the branch will be given by the number of leaves above it.

Productions p_3 through p_6 define the behavior of the apex. The first parameter of the module A is a counter that defines the bifurcation rate of the apex. It is used also to switch between leaf and flower production. When the counter reaches zero, production p_3 bifurcates the apex into two new branches. Production p_4 generates a flower if the tree has more than 5 years $(t > 5)$, if the apex counter is even $(x \bmod 2 = 0)$ and if it is spring. Production p_5 generates a leaf if the apex counter is odd $(x \bmod 2 = 1)$ and if we are between the end of the winter and the end of the summer. Production p_6 takes place if none of the above productions were matched and decreases the apex counter.

The growth of the flowers is defined by productions p_7 through p_9. The flower is generated in production p_7: the module $@v$ rolls the drawing turtle so that the flower is drawn in the correct angle, the module $,(1)$ sets the current color to purple, and the module $\sim flower$ draws the flower. Production p_8 decreases the flower counter and production p_9 uses the special module $\%$ that crops the flower when the counter reaches zero.

Productions p_{10} through p_{13} defines the growth of the leaves. The leaf is generated in production p_{10}: the module $@v$ rolls the drawing turtle so that the leaf is drawn in the correct angle, the module $@D(\sqrt{t})$ sets the size of the leaf according to the age of the tree, the module $,(2)$ sets the current color to green, and the module $\sim leaf(0, 0)$ draws the leaf. The leaf module has two parameters because it is the source module of the photosynthesis signal. Production p_{11} decreases the leaf counter and production p_{12} removes the leaf when the counter reaches zero. Production p_{13} is responsible for changing the color of the leaves in the fall. The color map of this model is set so that the colors 3 through 6 change gradually from green to orange.

Figure 9 shows the development of this model. The internodes were drawn using generalized cylinders [Pru96a] instead of line segments so that the image generated is more realistic.

This example demonstrates how easy it is to build a complex model with many signals using the presented technique. As the L-System captures only the reaction of the modules reached by a relevant signal, the productions are simple and objective. For example, the nutrient concentration signal is used only in production p_1 where the internode length is set, and the photosynthesis signal is used only in production p_2 to define the internode width. In a model build without the presented technique, production p_1 would have to describe the signals propagation together with the internode elongation. Furthermore, the photosynthesis model utilized would have to be adapted because it is almost impossible to model a signal that is generated in the top of the tree and reaches the base of the tree in the next derivation.

This example also illustrates the independence of the signals. The model has four different signals, each one with its own velocity and propagation function. The propagation of one signal does not affect the other signals.

The photosynthesis signal shows a signal with many source modules. The value of the implicit parameter of a module reached by more than one signal of the same type is given by the sum of the signal functions calculated for that module. In this example, the implicit parameter will be equal to the number of leaves above the module. One interesting aspect of the this signal is that the velocity of the propagation is set to zero ($\sim leaf(0, 0)$). As the propagation function is a constant value (does not depend on the x coordinate), all the modules in the straight path from the source module to the root will have the same signal's value, independently of the velocity of propagation.

Another interesting aspect of this model is that it is a context free L-System. The presented technique makes the context sensitive L-Systems more rare because the contexts were used mainly to simulate the signal flow. In spite of that, the context sensitiveness is still useful to describe the reaction of the modules subject to their neighbors.

6 MODELING ENVIRONMENT

The technique presented in this paper was implemented in a software called LSLab which we developed using C++ with OpenGL for the Windows operating system. This software is available for download at:

`http://cgcap.ime.usp.br/lslab`

The LSLab was inspired on the cpfg software developed by Prusinkiewicz [Mec98a] which is currently the most complete and reliable L-System modeler software. We have decided to mimic the cpfg modeling language to make it possible to use the same model in both programs and because this is the most common notation for L-Systems.

7 CONCLUSION AND FUTURE WORK

In this paper we illustrated the benefits of the global signal propagation technique with relation to the previous methods for modeling signal and nutrient propagation in plant models. This new technique made the signal modeling a much easier and intuitive task. The improvements observed are:

- Full control over the velocity of the signal propagation: the velocity of the propagation is defined by a parameter of the signal source module.
- The signal respects the length of the modules traversed: the propagation of the signal is given by a function that sets the value of the signal for each module according to its distance from the source module.
- The signal can propagate through many modules in one derivation.
- It is easy to aggregate new signals into existing models: as the productions do not need to describe the propagation of the signal, it is easy to add a new signal to an existing model and then add or change some productions to describe the modules reaction to the new signal.
- A model can have many different signals and their propagations are completely independent.
- The L-System is clear and objective: as the productions only describe the behavior of modules reached by a signal, they are simple and objective.

Further research could be done to add stochastic rules to the signal description: the velocity and the propagation function could be randomly altered to capture specimen variation. This technique should also be tested in open L-Systems [Mec96b] where the propagation function could be set and changed by the environment to simulate the change in the supply of some nutrients. Another improvement of this technique could be done by making the velocity of the propagation vary depending on the module it traverses. The intensity of the signal could also vary this way.

In summary, we believe that the proposed signal modeling technique will prove useful in many aspects of plant development simulation.

8 REFERENCES

[Deu98a] O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 275–286. ACM Press, 1998.

[Fol95a] G. B. Folland. Introduction to Partial Differential Equations. In *Princeton University Press*, 1995.

[Fow92a] D. R. Fowler, H. Meinhardt, and P. Prusinkiewicz. Modeling seashells. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 379–387. ACM Press, 1992.

[Ger96a] M. Gervautz and C. Traxler. Representation and realistic rendering of natural phenomena with cyclic csg graphs. In *The Visual Computer*, volume 12, pages 62–74. Springer-Verlag New York, Inc., 1996.

[Mec98a] R. Mech. *CPFG Version 3.4 User's Manual*, 1998.

[Mec96b] R. Mech and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 397–410. ACM Press, 1996.

[Par01a] Y. I. H. Parish and P. Mller. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308. ACM Press, 2001.

[Pru96a] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Mech. Visual models of plant development. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume 3, chapter 9, pages 535–597. Springer-Verlag, 1996.

[Pru93b] P. Prusinkiewicz, M. S. Hammel, and E. Mjolsness. Animation of plant development. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 351–360. ACM Press, 1993.

[Pru94c] P. Prusinkiewicz, M. James, and R. Mech. Synthetic topiary. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM Press, 1994.

[Pru90d] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., 1990.

[Die97a] D. Schmalstieg and M. Gervautz. Modeling and rendering of outdoor scenes for distributed virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 209–215. ACM Press, 1997.

[Tra97a] C. Traxler and M. Gervautz. Efficient raytracing of complex natural scenes. In *Proceedings Fractals'97, World Scientific Publishers*, 1997.

Figure 9: Two years of the development of the tree from Example 3. The images begin in the winter. The flowers appear in the spring and disappear in the middle of the summer. In the fall, the leaves become orange and fall, and, until near the end of the winter, the tree remains with no leaves. The figure shows some stages of the development from derivation 166 to 204.