

Animating Plant Growth in L-System by Parametric Functional Symbols

Yodthong Rodkaew,^{1,*} Somporn Chuai-aree,^{2,†} Suchada Siripant,^{2,‡}
Chidchanok Lursinsap,^{2,§} Prabhas Chongstitvatana^{1,¶}

¹*Department of Computer Engineering, Chulalongkorn University,
Bangkok, Thailand 10330*

²*Department of Mathematics and Computer Science,
Chulalongkorn University, Bangkok, Thailand 10330*

This article proposes an animating system for plant growth in the Lindenmayer (L) system in which alphabets are parameterized to the length, size, and position of each component of the plant. The growth rate is approximated by sigmoidal curves. To achieve a realistic leaf shape, a set of shape functions is used to transform a skeleton of primary branches into the final shape. Genetic algorithm (GA) is used to fit the shape's parameters. Modeling of the soybean is shown. The developments of plant growth are smooth and natural as well as realistic. This prototype can be used to generate a realistic model of any plant based on the L-system. © 2004 Wiley Periodicals, Inc.

1. INTRODUCTION

Time-lapse photography reveals the enormous visual appeal of developing plants, related to the extensive change in topology and geometry during growth. Consequently, the animation of plant development represents an attractive and challenging problem for computer graphics.¹⁻⁶

This research presents a prototype for creating computer models that capture the development of plants using the Lindenmayer (L) system and mathematical model incorporating biological data. The L-system is used for qualitative model in order to represent plant topology and development.⁷ There are six consecutive steps in this method: (1) defining a qualitative model constructed from observations of plant growth in their life cycle, (2) measurement of key characteristics collected

*Author to whom all correspondence should be addressed: e-mail: yodthong.r@Student.chula.ac.th.

†e-mail: csomporn@math.sc.chula.ac.th.

‡e-mail: Suchada.S@Chula.ac.th.

§e-mail: lchidcha@chula.ac.th.

¶e-mail: prabhas@chula.ac.th.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 19, 9-23 (2004)

© 2004 Wiley Periodicals, Inc. Published online in Wiley InterScience
(www.interscience.wiley.com). • DOI 10.1002/int.10147

from actual plants, (3) converting raw data to growth functions based on sigmoid function approximations, (4) defining a quantitative model composed from the qualitative model and growth function, (5) visualization of the quantitative model, and (6) model evaluations.

The visualization aids in exposing any flaws in the qualitative or quantitative models and helps to identify any incorrectly estimated functions.⁸⁻¹² This research uses soybean (*Glycine max.*) for a case study.

The rest of this study is organized into six sections. Section 2 summarizes the concept of a general L-system. Section 3 expresses the plant modeling, the qualitative model of plant structure, and the growth function approximation from the actual plant. Section 4 describes a method to generate a realistic leaf shape. Section 5 shows the visualization and discusses model evaluation. The conclusion is given in Section 6.

2. L-SYSTEM

The L-system was first introduced in 1968 by Aristid Lindenmayer as a mathematical theory of plant development. It attracted the attention of the computer scientists who investigated it through formal language theory. Specialists in computer graphics, particularly Prusinkiewicz, have used the L-system to produce realistic image of trees, bushes, and flowers, and some of images are well illustrated in *The Algorithmic Beauty of Plants*.¹

The L-system is formal grammar that is characterized by having a single axiom, no terminal symbols, and performing all possible symbol replacements in parallel for each step. A parametric 0L-system, which are context-free, operate on *parametric words*, which are a string consisting of *letters* and *parameters* called *modules*. The letter as in the alphabet is denoted by V , and the set of parameters is the set of the real numbers \mathfrak{R} . A module with letter $A \in V$ and parameters $a_1, a_2, \dots, a_n \in \mathfrak{R}$ is denoted by $A(a_1, a_2, \dots, a_n)$. Every module belongs to the set $M = V \times \mathfrak{R}^*$, where \mathfrak{R}^* is the set of finite sequences of parameters. The set of all the strings of modules are denoted by $M^* = (V \times \mathfrak{R}^*)$. The set of all nonempty strings are denoted by $M^+ = (V \times \mathfrak{R}^*)^+$.

The real-valued *actual* parameters appearing in the words correspond with *formal* parameters, which may occur in the specification of the L-system productions. Let Σ be a set of formal parameters, $C(\Sigma)$ denotes a *logical expression* with parameter from Σ , and $\varepsilon(\Sigma)$ is an *arithmetic expression* with parameter from Σ .

A parametric 0L-system is defined as an ordered quadruple $G = (V, \Sigma, \omega, P)$ where

- V is the alphabet of the system
- Σ is the set of formal parameters
- $\omega \in (V \times \mathfrak{R}^*)^+$ is a nonempty parametric word called the axiom
- $P \subset (V \times \Sigma^*) \times C(\Sigma) \times (V \times \varepsilon(\Sigma))^*$ is a finite set of productions.

For example, if the alphabets V in the system are $\{F, +\}$, which may occur many times in a string. Each letter is associated with a rewriting rule. The rule $F \rightarrow F +$

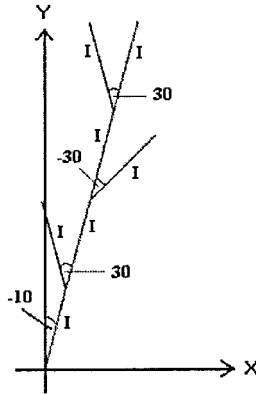


Figure 1. Example of a simple two-dimensional space L-system, $I[+I]I[-I]I[+I]I$.

F means that letter F is to be replaced by $F + F$. The rewriting process starts from a distinguished string called the *axiom* or ω . Given the axiom string F , in the first derivation step, the string F is replaced by string $F + F$ to be string $F + F$. In the second derivation step, the string $F + F$ is replaced by string $F + F + F + F$.

A simple L-system is based on turtle graphics. The basic idea of turtle interpretation described by Prusinkiewicz, Hammel and Mjolsness³ is as follows. A turtle can move in any direction, either forward, backward, rightward, or leftward. Each movement is defined using three primitive symbols, namely, I , $+$, and $-$. Symbol I denotes one unit length of movement, which can be in any direction. The direction is defined by symbols $+$ and $-$. Symbol $+$ denotes movement in a counter-clockwise direction and symbol $-$ denotes the movement in a clockwise direction. A branch is defined by symbols $[$ and $]$. Symbol $[$ denotes the beginning of the branch and symbol $]$ denotes the end of branch. The movement can be transformed to a Cartesian coordinate system described by a triplet (X, Y, α) , where X and Y indicate the coordinate of the movement and α indicates the initial angle of the first unit movement with respect to the Y -axis. The α can be either positive or negative. The other unit movement can have its directional angles computed by adding or subtracting a constant δ to the initial angle α . Similarly to the initial angle α , the computed angle is rotated with respect to the Y -axis. Figure 1 shows an example of how symbols I , $+$, $-$, $[$, $]$ are used to describe the movement of a turtle with the XY -coordinates and the directional angle α . Here, we set α to -10° and δ to 30° .

In case of a three-dimensional movement, a turtle is free to move in any X -, Y -, or Z -direction. Hence, the directional angles in this case become three. Initial three directional angles α_x , α_y , and α_z of the first unit movement are set with respect to X -, Y -, and Z -axes, respectively. The directional angles of the other unit movements are computed in a similar fashion to that of the two-dimensional case. Three constants, δ_x , δ_y , and δ_z , are used to adjust the direction of the unit movements. In addition, the physical location of the unit movement I previously defined must be represented an XYZ -coordinates. Therefore, a unit movement

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 2. Rotation matrices.

described in the Cartesian coordinate system is denoted by a hexaplet $(X, Y, Z, \alpha_x, \alpha_y, \alpha_z)$. After adding/subtracting δ_x , δ_y , and δ_z , the new XYZ-coordinates of the movement are computed by multiplying the coordinates of the current movement with the rotation matrices \mathbf{R}_x , \mathbf{R}_y , and \mathbf{R}_z shown in Figure 2. The rotation of a unit movement and its direction are captured in a symbolic form similar to that in the two-dimensional case by using these symbols $/, \backslash, \&, ^, +, -, \text{ and } |$. The meaning of each symbol is explained in Table I. An example of a simple turtle in a three-dimensional space is

$$(\delta_x = \delta_y = \delta_z = 70, \alpha_x = \alpha_y = \alpha_z = 0) \quad I[-I][+I][/I][\backslash I]I$$

We define the symbol I as an internode of a plant. The example has 10 internodes, 6 internodes for main stem, and 4 internodes of petiole in each direction. The visualized image is shown in Figure 3.

Table I. Symbols used in the plant growth L-system.

Symbols	Meaning
I	It is used to generate the plant internodes
i	It is used to generate the plant short internodes
P	It is used to generate the plant petioles
p	It is used to generate the plant short petioles
A	It is used to generate the plant apices
L	It is used to generate the plant leaves
F	It is used to generate the plant flowers
$+$	Turn left by angle δ_z , using rotation matrix $\mathbf{R}_z(\delta_z)$
$-$	Turn right by angle δ_z , using rotation matrix $\mathbf{R}_z(-\delta_z)$
$\&$	Pitch down by angle δ_y , using rotation matrix $\mathbf{R}_y(\delta_y)$
\wedge	Pitch up by angle δ_y , using rotation matrix $\mathbf{R}_y(-\delta_y)$
\backslash	Roll left by angle δ_x , using rotation matrix $\mathbf{R}_x(\delta_x)$
$/$	Roll right by angle δ_x , using rotation matrix $\mathbf{R}_x(-\delta_x)$
$ $	Turn around, using rotation matrix $\mathbf{R}_x(180)$
$[$	Push the current state to stack; it is used to set the degree of the plant branch
$]$	Pop the current state from stack

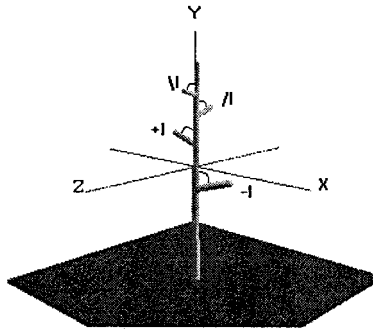


Figure 3. A simple L-system interpretation.

3. PLANT MODELING

The L-system description of a plant is defined in forms of a set of iterations, a set of directional and sizing parameters, an initial string, a set of production rules, and a set of terminating productions.

The angle (δ) is used to set the angle of a the branch. For example, “-” is to turn right by an angle δ , “^” is used to pitch up by an angle δ , and “/” is used to roll right by an angle δ . The diameter (D) is used to set the diameter of the first internode.

The axiom (ω) is used to set the start status of the plant. Every start stem is located at the origin $(0, 0, 0)$ and pointed toward the positive Y -axis. The three angles for a three-dimensional space ($\alpha_x, \alpha_y, \alpha_z$) are set to zero for the first internode.

The production rule (γ), called “endrule,” is used to terminate the rewriting of a production. A terminating symbol must be substituted for the corresponding symbol used by the previously called production rule.

3.1. Qualitative Model

The modeling process begins with the specification of the qualitative model. It captures the aspects of a plant that can be obtained through the observations and are deemed essential to its form and development. These include the topology and the sequence of activities of various plant modules. The main components of the plant are distinguished and their developmental stages are identified. The connections between these components also are defined. In this article, we obtain the qualitative parameters from a soybean.

The qualitative parameters of the soybean model consist of three main parts: internodes, petioles, and leaves. The simulation begins with first pair of leaves. This is captured by the L-system axiom or the initial string of modules. The axiom in Figure 4 represents an internode, a pair of leaves and its short internode, and an apex. The apex A is contained initially within the petioles. After some iterations, the apex A is substituted by an internode I , a right petiole P , an internode I , a left petiole B , and an apex A shown by the following production rule:

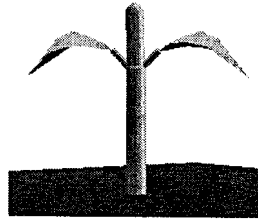


Figure 4. Axiom = $I[+iL][-iL]A$.

$$A \rightarrow I[-P]I[+B]A$$

Each petiole consists of some internodes, some short petioles, a left leaf, a right leaf, and a middle leaf. The production rules on the left and right petioles are defined as follows.

$$P \rightarrow IIII[\backslash pL][/pL][-pL]$$

$$B \rightarrow IIII[\backslash pL][/pL][+pL]$$

The “endrule” is defined as follows.

$$B \rightarrow IL$$

$$P \rightarrow IL$$

$$A \rightarrow IL$$

The foregoing endrules are called at the last iteration. The left petiole (B) as well as the right petiole (P) and the apex (A) are substituted by an internode and a leaf.

By using the previously defined production rules with some specific parameters, the L-system description of a soybean is given as

$$\begin{aligned} \omega & I[+iL][-iL]A \\ P_1 & A \rightarrow I[-P]I[+B]A \\ P_2 & P \rightarrow IIII[\backslash pL][/pL][-pL] \\ P_3 & B \rightarrow IIII[\backslash pL][/pL][+pL] \\ \gamma_1 & A \rightarrow IL \\ \gamma_2 & P \rightarrow IL \\ \gamma_3 & B \rightarrow IL \end{aligned}$$

where iterations = 6, angle (δ) = 45.0° , and diameter (D) = 1.5 mm.

In the foregoing L-system, the number of iterations is six; the initial branch angle is 45° and the diameter of first internode is 1.5 mm. After the sixth iteration, the endrule productions are called to terminate the substitution process.

3.2. Data Collection

The data of each component are collected from an actual soybean plant. They are the internode length, diameter, leaf length and width, flowers, and petiole

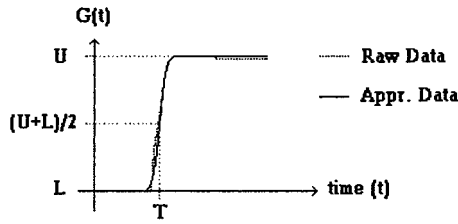


Figure 5. Sigmoidal curve approximation.

length corresponding to the time of its life cycle. The actual data are obtained from five soybeans everyday for 61 days. The raw data will be used for approximating the sigmoidal growth function. The data of the soybean are collected manually using rulers and a protractor.

3.3. Growth Function

The raw data in Section 3.2 is approximated as a growth function in Figure 5 using a sigmoidal curve approximation. The raw data is converted to growth function $G(t)$ of length or width at time t and is as

$$G(t) = L + \frac{U - L}{1 + e^{m(T-t)}}$$

where

- L the minimum value of length or width
- U the maximum value of length or width
- m the approximated slope of raw data
- T the time at $(U + L)/2$
- t the independent time variable

Besides the growth function, there is another function that we use to control all the components of the plant topology, such as the length of each internode from the first node to the last node. The function is

$$Y_i = c \cdot a^{n_i}$$

where Y_i is the length of node i , c is a constant, a is a real value greater than zero, and n_i is the level of node i .

The initial time of each component is specified by the following linear equation:

$$B_i = \beta n_i + b$$

where B_i is the initial time of component i , β is the acceleration rate of B_i , n_i is the level of component i , and b is a constant.

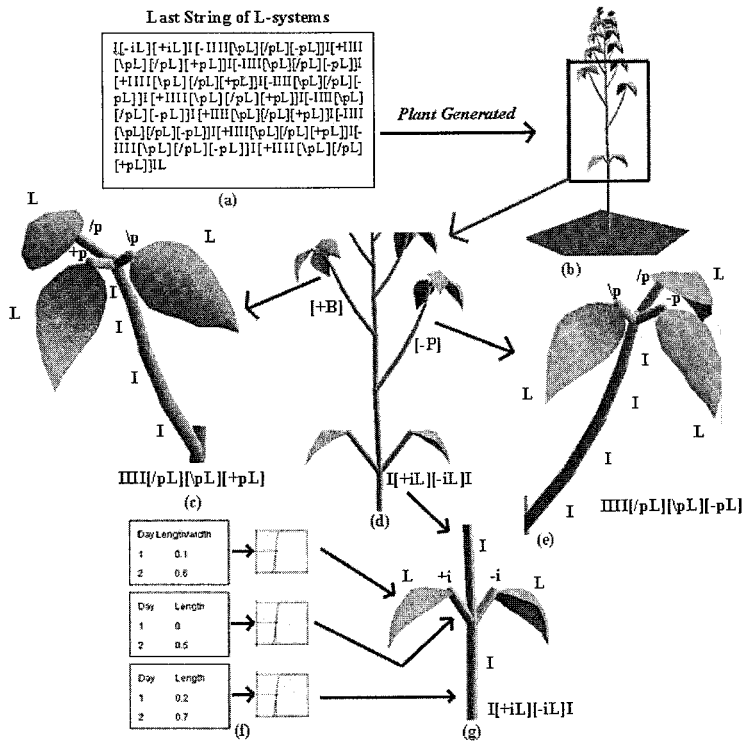


Figure 6. Structure of simulation generated from the production rules of the soybean in Section 3.

Every component of a plant is controlled by a self-growth function with either the same or different slope m , time T , the maximum U , and the minimum L .

Figure 6 shows a structure of simulation generated from the production rules of the soybean. Figure 6(a) shows the string of the L-system after rewriting the process. The plant structure in Figure 6(b) is constructed from that L-system string. Figure 6(d) shows the details of a part of a soybean. Figure 6(c) is the left petiole component. The right petiole is shown in Figure 6(e). Figure 6(g) shows the axiom of the soybean. The raw data are converted to growth function corresponding to each symbol like Figure 6(f).

4. LEAF SHAPE

Generally, there are some parts of a plant that are difficult to derive by a rewriting expression such as leaves or flowers. Finding a rewriting expression by hand for a realistic leaf shape is too difficult. Here, the technique proposes writing a grammar for the primary branch and using a genetic algorithm (GA) to fit the shape's parameters.¹³

The target picture is taken from the outline of a real leaf. A sample of the target is shown in Figure 7.

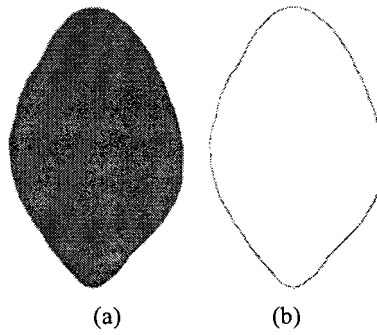


Figure 7. (a) The real leaf and (b) its outline.

4.1. Leaf Expression

The leaf shape is derived from a skeleton. The following grammar describes a skeleton:

ω [BBBBBBBBBB] [CCCCCCCCC]
 P_1 $B \rightarrow J[-gU]$
 P_2 $C \rightarrow J[+hV]$
 P_3 $U \rightarrow qKKU$
 P_4 $V \rightarrow rKKV$
 γ_1 $U \rightarrow$
 γ_2 $V \rightarrow$

where iterations = 8 and angle (δ) = 90.0°. The meaning of each symbol is explained in Table II.

4.2. Shape Function

To shape a skeleton, the turning angle g , h , q , and r as well as the size K are parameterized according to a set of shape functions. These functions are described by β -spline in Figure 8. Each segment is parameterized by (m, n) , where m is an

Table II. Symbols used in the leaf shape L-system.

Symbols	Meaning
B, C	A branch
U, V	Segments of a branch
J	The segment between branches along the length of a leaf
K	The length of a segment of a branch
g, h	Initial angle of a segment
q, r	Turning angle of a segment

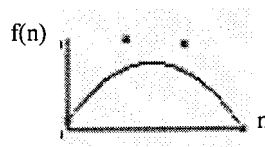


Figure 8. A β -spline function.

index along the length of a leaf and n is an index along the width. The n is the number of derivations (iteration number).

To vary the shape smoothly, β -spline functions are fitted to g , h , q , and r . Because q and r vary both with m and n , two splines are mixed with the following formula:

$$\theta = m \times S_1 + (1 - m) \times S_2$$

where $m \in 0, 1$; S_1 and S_2 are two splines indexed by n ; θ is the turning angle; and J , g , and h each associates with a spline indexed by n .

Figure 9 shows various shapes that can be achieved using different sets of splines.

4.3. GAs

GAs¹⁴ are used to find a set of shape functions that generated primary branches that fit an outline of a real leaf (Figure 7). GA is based on an inspiration from natural selection. It was developed by John Holland, his colleagues, and his student at the University of Michigan. It is a robust algorithm for the searching and optimization of solutions. Each solution is called an *individual*. A group of

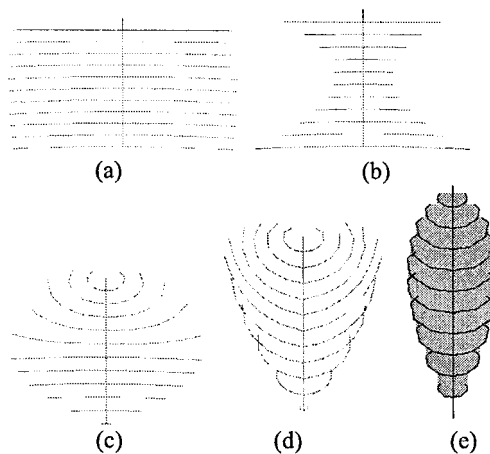


Figure 9. The picture created from rules: (a) the skeleton and (b)–(d) various transformation shaped by shape function. (e) An example of a leaf with hand-written shape function.

individuals are called a *population*. GA evaluates each individual to measure *fitness*. An individual who has high fitness can produce their children or *offspring* for the next *generation*.

In the first generation, GA randomly generates a population and evaluates them. In the second generation, GA selects an individual with probability according to its fitness to produce offspring and modifies them by *genetic operators* (reproduction, crossover, and mutation). This process is repeated until the terminating condition is met, such as an individual can solve the problem or the specific number of generations has passed.

4.3.1. Individual

An individual in this experiment denotes the parameters of the β -spline function. Each spline has four parameters. There are five splines in this experiment; therefore, an individual contains 20 floating-point variables.

4.3.2. Genetic Operator

The *reproduction*, *crossover*, and *mutation* are used to find a solution. The reproduction is the duplication from the parent to their child. The crossover operator selects two individuals and splits each individual into two parts at a random point and recombines them. This method generates two offstrings. The mutation produces an offspring by randomly changing four values in the parent.

4.3.3. Fitness Function

The fitness for each shape function is the error of the generated leaf from the target outline.

4.3.4. Genetic Parameters

The genetic parameters are as follows:

Number of individuals	200
Reproduction	20%
Crossover	40%
Mutation	40%
Number of generations	200

4.4. Leaf Shape Results

Figure 10 shows the results of the final generation. The output matches closely to the target. Figure 11 shows the simulation of smooth growth of leaf shape.

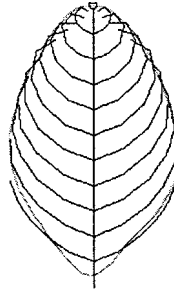


Figure 10. The leaf development at the 200th generation.

5. VISUALIZATION

The visualization of the soybean is shown in the vegetative state. Cylinders are used to represent internode and petiole segments. Spheres are used to represent jointed internodes. Triangle polygons are used to represent leaves and flowers. Figure 12 shows some selected stages of the development of a soybean shoot controlled by the production rules defined in Section 3. The developments in Figure 11 are started at time $t = 20$ according to the sigmoidal curve in Figure 5.

Figure 8 shows different plant structures with the same topology of the L-system under different parameters. The L-system of Figure 8 is

$$\begin{aligned}
 \omega & I[-1][+1][\backslash 2][\wedge 2][^1][^1][^{\backslash -1}][^{\backslash -1}][^2][^2] \\
 P_1 & 1 \rightarrow I[iL][iL]1 \\
 P_2 & 2 \rightarrow I[-iL][+iL]2 \\
 \gamma_1 & 1 \rightarrow IF \\
 \gamma_2 & 2 \rightarrow IF
 \end{aligned}$$

where iterations = 8, angle (δ) = 1.5° , and diameter (D) = 0.8 mm.

In Figure 13, we use eight iterations to four plants. Each plant looks like a different species. The symbol L and F are linked from our leaf and flower library.

If there are any flaws present in the plant model, the visualization will make them clear. The adjustment of appropriate parameters such as angle, size, length, and width can alter the appearance of the output.

It is easy to modify the plant model to make it more realistic by interacting with the visualization system.

6. CONCLUSIONS

This work presents a prototype system to animate plant growth including leaf shapes. It is useful for creating the continuous development of plant models. The main contribution is the technique of parametric functional symbols based on the bracketed L-system. A case is presented using the soybean to validate the prototype.

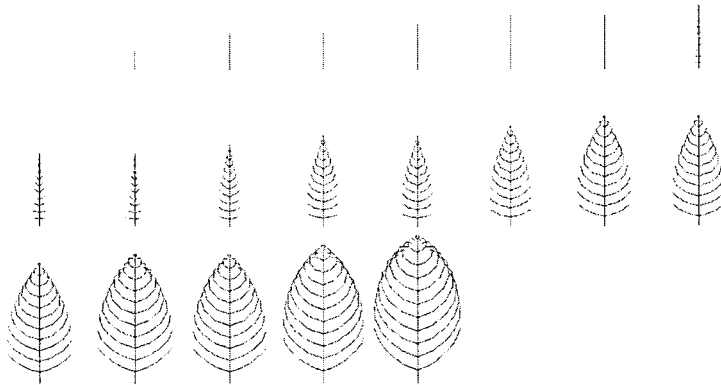


Figure 11. Simulation of the smooth growth of leaf shape.

Tong Lin¹⁵ implemented the animation of the L-system in a three-dimensional space plant growing in Java. He developed the plant growth animation by iterating the L-system. The work presented here differs in the use of parametric functional symbols, which make the animation smoother. The approach to generate leaf shapes based on parameter fitting by GAs offers flexibility in generating many realistic leaf shapes.

This prototype can be used to generate the realistic model of any plant that has a life cycle similar to the soybean.

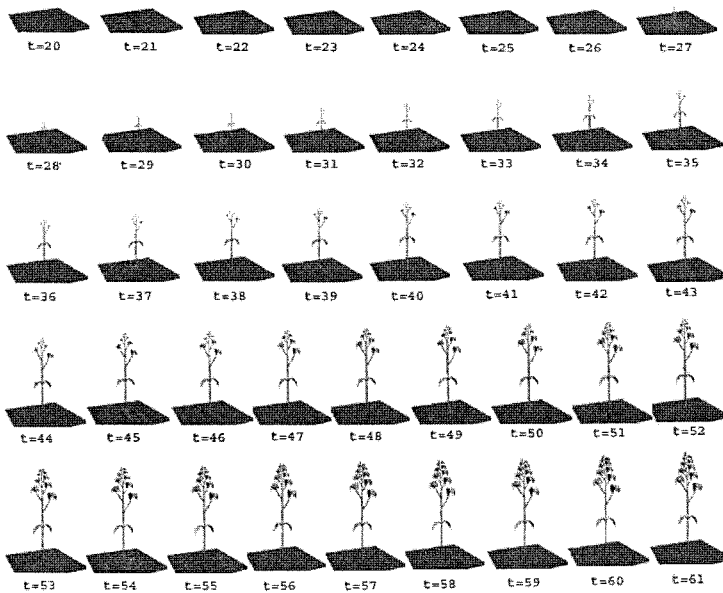


Figure 12. Simulation and visualization of soybean shoot expansion over 61 days.

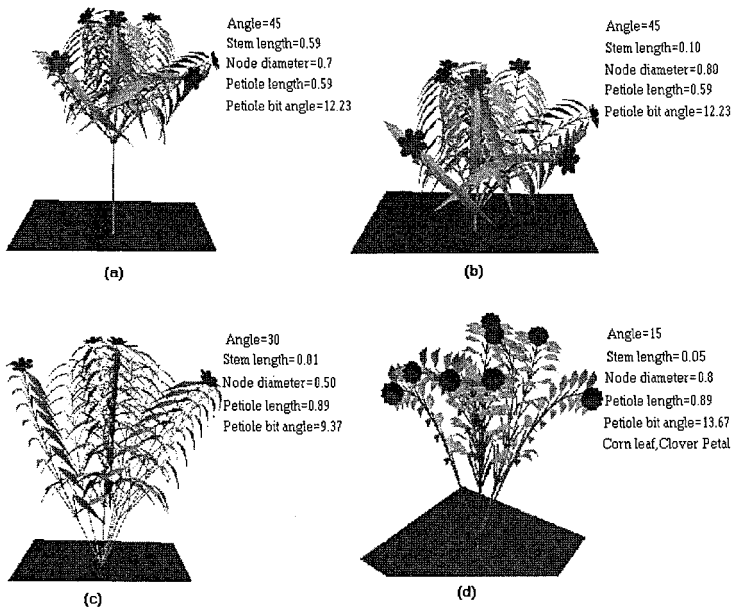


Figure 13. Different parameters of the same topology of the L-system.

Acknowledgments

We would like to thank Jim Hanan for much information, suggestions, and discussions in this research and Przemyslaw Prusinkiewicz who developed the L-studio program that we used to study the L-system and animation of plant model. This work is partially supported by a grant from the TJTTP project, which gives a collaboration with Prof. Shinagawa of the University of Tokyo.

References

1. Prusinkiewicz P, Lindenmayer A. The algorithmic beauty of plants, Vol 1. New York: Springer-Verlag; 1990.
2. Mech R, Prusinkiewicz P. Visual models of plants interacting with their environment. In: Proc Computer Graphics, SIGGRAPH '96 Proceedings. 1996. pp 397–410.
3. Prusinkiewicz P, Hammel MS, Mjolsness E. Animation of plant development. In: SIGGRAPH Computer Graphics Annual Conference, Vol 27, 1993. pp 351–360.
4. Prusinkiewicz P, Hammel M, Hanan J, Mech R. In: Visual model of plant development. In: G Rozenberg and A Salomaa, editors. pp 535–597. Handbook of formal language. New York: Springer-Verlag; 1996.
5. Hanan J. Virtual plants—integrating architectural and physiological plant models. In: Proc ModSim 95, Vol 1, The Modelling and Simulation Society of Australia. Perth; 1995. pp 44–50.
6. Prusinkiewicz P, James M, Mech R. Synthetic topiary. In: Proc of SIGGRAPH '94, Orlando, Florida, July 24–29, 1994. ACM SIGGRAPH. New York; 1994. pp 351–358.
7. Prusinkiewicz P, Kari L. Subapical bracketed L-system. In: Proc 5th Intl Workshop on Graph Grammars and their Application to Computer Science, Williamsburg, 1994.

8. Room PM, Hanan JS. Virtual cotton: A new tool for research, management, and training. In: Proc World Cotton Research Conf, Brisbane, Australia, February 14–17, 1994. pp 40–44.
9. Room PM, Hanan JS, Prusinkiewicz P. Virtual plants: New perspectives for ecologists, pathologists, and agricultural scientists. *Trends Plant Sci* 1996;1:33–38.
10. Room PM, Maillette L, Hanan J. Module and metamer dynamics and virtual plants. *Adv Ecol Res* 1994;25:105–157.
11. Lintermann B, Deussen O. A modelling method and user interface for creating plants. Proc Graphics Interface, Kelowna BC Mai 1997, Morgan Kaufmann Publishers. The Eurographics Association; 1998.
12. Lintermann B, Deussen O. Interactive modeling of plants. *IEEE Comput Graph Appl*, Vol 19, 1999.
13. Lursinsap C, Sophatsathit P, Siripant S. Simulation of leaf growth based on a rewriting system: A unified leaf model. Advanced Virtual and Intelligence Computing (AVIC) Research Center, Chulalongkorn University, Technical Report No. 01.01.2000.
14. Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Redwood City, CA: Addison-Wesley; 1989.
15. Lin T. Animation of L-system based 3-D plant growing in Java. CMSC-693, Research and Writing Skills, 1996.