

# Smooth Animation for Plant Growth Using Time Embedded Component and Growth Function

Somporn Chuai-Aree<sup>1</sup>, Willi Jäger<sup>1</sup>, Hans Georg Bock<sup>1</sup>, Suchada Siripant<sup>2</sup>

<sup>1</sup>*Interdisciplinary Center for Scientific Computing (IWR)  
University of Heidelberg, INF 368, D-69120 Heidelberg, Germany  
{Somporn.ChuaiAree, wjaeger, bock}@iwr.uni-heidelberg.de*

<sup>2</sup>*Advance Virtual and Intelligent Computing Center (AVIC)  
Chulalongkorn University, Phayathai road, Bangkok 10330, Thailand  
Suchada.S@chula.ac.th*

---

**ABSTRACT-** The iteration of the Lindenmayer system (L-system) has been used to animate the plant growth, however the development of the plant model at each time step is not smooth and continuous. This paper proposes an animating plant growth in L-system by a time embedded component and growth function for the length, size and position of each component of the plant shoot and root, such as, plant internodes, leaves, flowers, petioles, apices and root parts. The development of plant growth seems to be smoother and more natural as well as realistic. This prototype can be used to generate the realistic model of any plant based on a Bracketed and Stochastic L-system.

**KEY WORDS** – Lindenmayer systems, realistic image, continuous simulation, visualization, plant growth, plant model, time embedded component, growth function

---

## 1. Introduction

The animation of plant growth represents an attractive and challenging problem for computer graphics. In the previous works, Tong Lin [9] implemented the animation of L-system in a 3-dimensional space plant growth in Java. Przemyslaw Prusinkiewicz *et. al.* [12] extended L-systems in a manner suitable for simulation to interact between a developing plant and its environment. They developed the plant growth animation by iterating the L-system, but the animation was not smooth. They solved this problem using time L-system and differential L-system [15].

This paper presents a prototype for creating computer models that capture the development of plants using L-systems and a mathematical model incorporating biological data. L-systems are used for a qualitative model in order to represent the plant topology and development. This paper also extends from our previous work in [18,19]. There are six consecutive steps in this method, namely, (1) defining a qualitative model from observations of plant growth in their life cycle, (2) measuring of key characteristics collected from actual plants, (3) converting raw data to growth functions based on sigmoidal curve approximations, (4) defining a quantitative model which is a combination between the qualitative model and growth function, (5) visualizing of the quantitative model and (6) evaluating the model. The visualization aids in exposing any flaws in the qualitative or quantitative models and helps to identify any incorrectly estimated functions.

The paper is organized into 7 sections. Section 2 summarizes the concept of a general L-systems. Section 3 describes plant module design. Section 4 expresses the growth function approximation. Section 5 shows the structure of plant simulation. Section 6 describes visualization. The conclusion is given in Section 7.

## 2. Lindenmayer Systems (L-systems)

Lindenmayer systems (L-systems) were first introduced by Aristid Lindenmayer in 1968 as a mathematical theory of plant development [14]. They have attracted the attention of the computer scientists who investigated

them through formal language theory. Specialists in computer graphics, particularly Prusinkiewicz have used the L-systems to produce realistic images of trees, bushes, flowers and some images are well illustrated in [14].

For a three dimensional movement, a component is free to move in any X, Y, or Z direction. Hence, the directional angles in this case are three directions. The initial three directional angles,  $\alpha_x, \alpha_y, \alpha_z$  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,  $\delta_x, \delta_y, \delta_z$ , are used to adjust the direction of the unit movements. In addition, the physical location of the unit movement must be represented by XYZ coordinates. Therefore, a unit movement described in the Cartesian coordinate system is denoted by a hexaplet  $(X, Y, Z, \alpha_x, \alpha_y, \alpha_z)$ . After adding/subtracting  $\delta_x, \delta_y, \delta_z$ , the new XYZ coordinates of the movement is computed by multiplying the coordinates of the current movement with the rotation matrices  $R_x, R_y, R_z$  shown in equation (1). The rotation of a unit movement and its direction are captured in a symbolic form similar to two dimensional case [18] by using these symbols /, \, &, ^, +, -, |. The meaning of each symbol is explained in Table 1.

Table 1. Symbols used in L-system prototype

Symbols	Meaning
I	To generate the plant shoot/root internodes
i	To generate the plant shoot/root short internodes
P	To generate the plant shoot/root petioles
p	To generate the plant shoot/root short petioles
A	To generate the plant shoot apices or root tip
L	To generate the plant leaves.
F	To generate the plant flowers.
+	Roll counterclockwise by angle $\delta_z$ , using rotation matrix $R_z(\delta_z)$
-	Roll clockwise by angle $\delta_z$ , using rotation matrix $R_z(-\delta_z)$
&	Roll counterclockwise by angle $\delta_y$ , using rotation matrix $R_y(\delta_y)$
^	Roll clockwise by angle $\delta_y$ , using rotation matrix $R_y(-\delta_y)$
\	Roll counterclockwise by angle $\delta_x$ , using rotation matrix $R_x(\delta_x)$
/	Roll clockwise by angle $\delta_x$ , using rotation matrix $R_x(-\delta_x)$
	Roll back, using rotation matrix $R_y(180)$
[	Push the current state of the turtle onto a pushdown stack to create a new branch
]	Pop a state from the stack and make it the current state of the turtle to close the branch

$$R_x(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\mathbf{q}) & -\sin(\mathbf{q}) \\ 0 & \sin(\mathbf{q}) & \cos(\mathbf{q}) \end{bmatrix}, R_y(\mathbf{q}) = \begin{bmatrix} \cos(\mathbf{q}) & 0 & -\sin(\mathbf{q}) \\ 0 & 1 & 0 \\ \sin(\mathbf{q}) & 0 & \cos(\mathbf{q}) \end{bmatrix}, R_z(\mathbf{q}) = \begin{bmatrix} \cos(\mathbf{q}) & \sin(\mathbf{q}) & 0 \\ -\sin(\mathbf{q}) & \cos(\mathbf{q}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \dots(1)$$

An example of a simple turtle in a 3-dimensional space is given below. ( $\delta_x = \delta_y = \delta_z = 45, \alpha_x = \alpha_y = \alpha_z = 0$ )

I[-IL]I[+IL]I[/IL]I[\IL]IIF

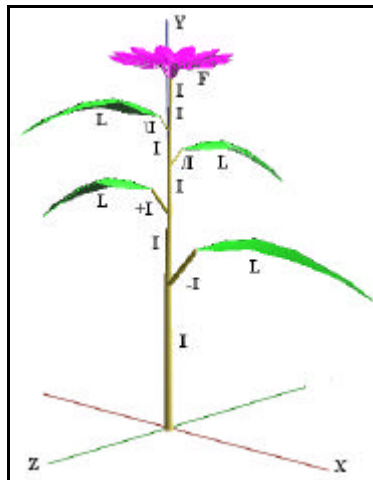


Figure 1. A simple L-system interpretation.

We define the symbol “*I*” for an internode, “*L*” for a leaf and “*F*” for a flower of the plant. From the example there are ten internodes - six internodes for the main stem, four internodes for the petiole and their leaves in each direction - and one flower. The visualized image is shown in Figure 1.

### Stochastic L-system

We use the Stochastic L-system in our system in order to generate variety of plant structure. The syntax of the production rule is defined as below.

$$\mathbf{Pred} = (\mathbf{Prob}_1) \mathbf{Succ}_1, (\mathbf{Prob}_2) \mathbf{Succ}_2, \dots, (\mathbf{Prob}_n) \mathbf{Succ}_n,$$

where **Pred** is the predecessor of successor **Succ<sub>i</sub>** with probability value **Prob<sub>i</sub>** when  $i=1,2,3,\dots,n$ . The sum of probabilities of all successors with the same predecessor **Pred** is equal to 1.

For example,  $I = (0.40) I[+iL]I, (0.30) I[-iL]I, (0.30) I[-iL][+iL]I$

The predecessor “*I*” will be replaced by three cases of successors with probability values 0.40, 0.30, 0.30, respectively.

### 3. Plant Module Design

The L-system description of a plant is defined in form of a set of iterations, a set of directional and a sizing parameters, an initial string, a set of production rules and a set of terminating productions. This description has the following format.

<pre> Plant_Name {     Shoot {         Iterations=N         Angle=δ         Diameter=D         Axiom=ω         Production 1         Production 2         ...         Production n         Endrule         Endproduction 1         Endproduction 2         ...         Endproduction m     } } </pre>	<pre> Root {     Iterations=N     Angle=δ     Diameter=D     Axiom=ω     Production 1     Production 2     ...     Production n     Endrule     Endproduction 1     Endproduction 2     ...     Endproduction m } </pre>
--	--

The meaning of each keyword is given as follows:

***Plant\_Name***

*Plant\_Name* is the name of the plant module. There are two modules of a plant prototype. They are “Shoot” and “Root”, respectively. The prototype of the shoot module is similar to the root module and uses the same keyword.

***Shoot, Root***

Shoot and Root are used to separate the shoot and root module, respectively.

***Iterations=N***

This input sets the number of iterations for selecting and rewriting the production rules. Each production rule is selected according to the appearance of the symbols in the current string. N is an integer greater than -1.

***Angle=d***

This angle(**d**) is used to set the angle of the branch. For example, ‘-’ is to turn right by an angle **d**, ‘^’ is to pitch up by an angle **d** and ‘/’ is to roll right by an angle **d**.

***Diameter=D***

This diameter is used to set the diameter of the first internode.

**Axiom =  $\mathbf{w}$** 

This string is used to set the start status of the plant. Every start stem is located at the origin (0,0,0) and pointed towards the positive Y axis. The three angles for a 3-dimensional space ( $\alpha_x, \alpha_y, \alpha_z$ ) are set to zero for the first internode.

**Production 1... Production n**

Each production consists of a predecessor and a successor. The format of production is given below.

**Predecessor = (Probability value) Successor**, the probability of normal production rule is equal to 1.

The predecessor is a symbol character and the successor is a symbol string.

**Endrule**

To terminate the rewriting of a production rule, a terminating symbol must be substituted for the corresponding symbol used by the endproduction rule. The substitution rules are defined in the *endproduction 1* to *endproduction m*. The *endproduction* rules are called at the  $N^{\text{th}}$  iteration.

**Endproduction 1 ... Endproduction m**

The format of endproduction rule is the same as the production rule. The endproduction is used to prevent some symbol that is not I, i, P, p, L, F or A.

**Character "{" and "}"**

The character "{" and "}" are the beginning and the ending of L-systems structure, respectively.

## 4. Growth Function

The raw data are approximated as a growth function in Figure 3 using a sigmoidal curve approximation.

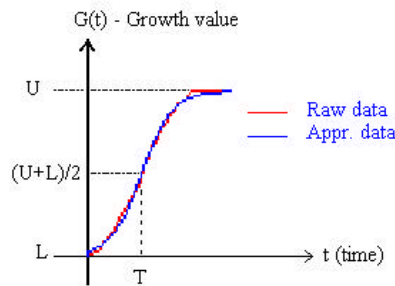


Figure 2. Sigmoidal curve approximation.

The raw data are converted to growth function  $G(t)$  of length or width at time  $t$  and is given below.

$$G(t) = L + \frac{U - L}{1 + e^{-m(T-t)}}, \quad \dots(2)$$

where L : the minimum value of length, width or diameter,  
 U : the maximum value of length width or diameter,  
 m : the approximated slope of raw data,  
 T : the time at  $G(t) = (U + L)/2$  and  
 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 to the last internode. The function is

$$Y_i = c(a)^{n_i}, \quad \dots(3)$$

where  $Y_i$  is the length of internode  $i$ ,  $c$  is a constant,  $a$  is a real value greater than zero and  $n_i$  is the level of internode  $i$ . We also use this equation to control the size of petiole, leaf, flower from the first to the last level.

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

$$B_i = \mathbf{b}n_i + b, \quad \dots(4)$$

where  $B_i$  is the initial time of component  $i$ ,  $\mathbf{b}$  is the acceleration rate of  $B_i$ ,  $n_i$  is the level of component  $i$  and  $b$  is a constant.

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

### 5. Structure of Plant Simulation

After measuring the plant, we have the plant structure as L-system prototype. There are two parts of L-system prototype. The first part is shoot prototype and the second part is root prototype. The L-strings of shoot/root part are generated by rewriting of shoot/root prototype (No. 1, No.3) and these L-strings represent the shoot/root structure (No. 2, No. 4) of the plant, respectively. These processes are given in Figure 3.

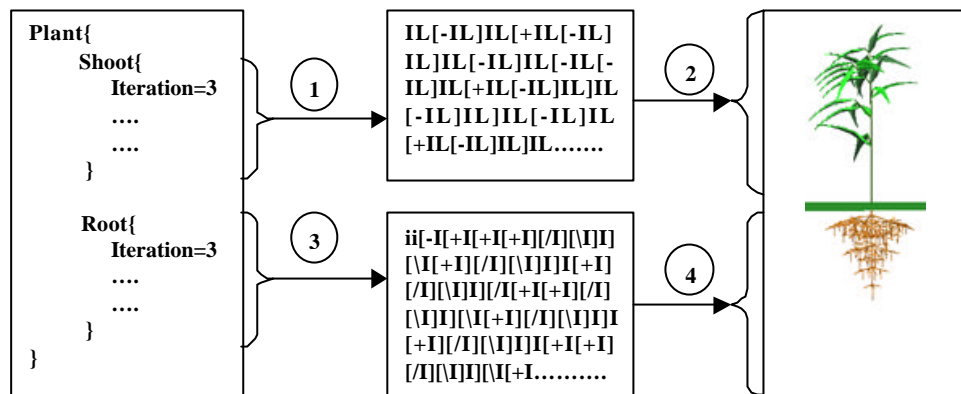


Figure 3. L-system transformation.

The structure of simulation starts from data collection of the actual plant and also captures the structure of the plant to be the L-system prototype. The measured raw data are approximated by the growth function based on a sigmoidal curve. The L-system prototype is compiled by the rewriting process to be an L-string at the last iteration. The L-String and the growth function of each symbol are combined together and represent the structure of the plant and its development. The system generates the 3D-plant development by visualization. We have to evaluate the plant growth and adjust some parameters, such as, the size of each internode, to make the plant model and plant growth to look more realistic. The structure of our simulation is shown in Figure 4.

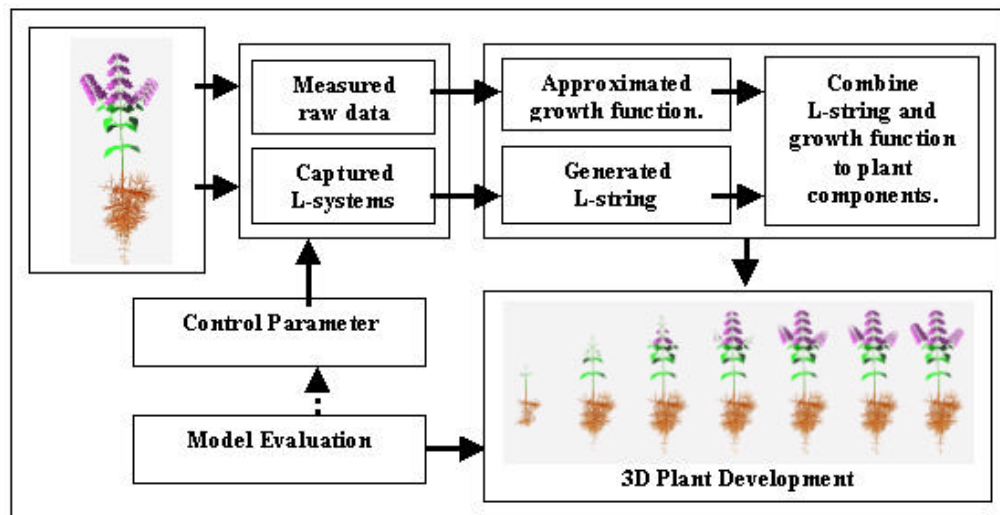


Figure 4. Structure of simulation and visualization of plant growth.

### 6. Visualization

In order to understand the plant structure using the Stochastic and Bracketed L-system, an example of plant structure is given below. There are three iterations of shoot part and root part. The shoot part has only one production rule with three cases of Stochastic L-system and one endproduction rule  $A=IL$ . For the root part there are four production rules using the Stochastic L-system, one production rule using normal rule and five endproduction rules A, B, C, D, E.

```

Plant{
  Shoot{
    Iterations=3
    Angle=45
    Diameter=0.7
    Axiom=A
    A=(0.33)A[+A]A[-A]A,(0.33)A[+A]A,(0.34)A[-A]A // 3 cases of Stochastic L-System
    Endrule
    A=IL // endproduction rule
  }
  Root{
    Iterations=3
    Angle=45
    Diameter=0.5
    Axiom=ii[-A][+B][\C][\D]E
    A=(0.5)I[+A][\A][\A]A,(0.25)I[+A][\A]A,(0.25)I[+A][\A]A // 3 cases of Stochastic L-System
    B=(0.5)I[-B][\B][\B]B,(0.25)I[-B][\B]B,(0.25)I[-B][\B]B // 3 cases of Stochastic L-System
    C=(0.5)I[-C][\C][\C]C,(0.25)I[-C][\C]C,(0.25)I[+C][\C]C // 3 cases of Stochastic L-System
    D=(0.5)I[-D][\D][\D]D,(0.25)I[+D][\D]D,(0.25)I[-D][\D]D // 3 cases of Stochastic L-System
    E=I[-A][+B][\C][\D]E // normal rule
    Endrule
    A=I // endproduction rule
    B=I // endproduction rule
    C=I // endproduction rule
    D=I // endproduction rule
    E=I // endproduction rule
  }
}

```

The above plant prototype is compiled and then we get the different sequences of the production rules. From the example there are 27 different structures ( $3^3$  different structures, 3 cases of stochastic for 3 iterations) of shoot part and  $3^{12}$  different structures (81 cases of stochastic for 3 iterations) of root part.

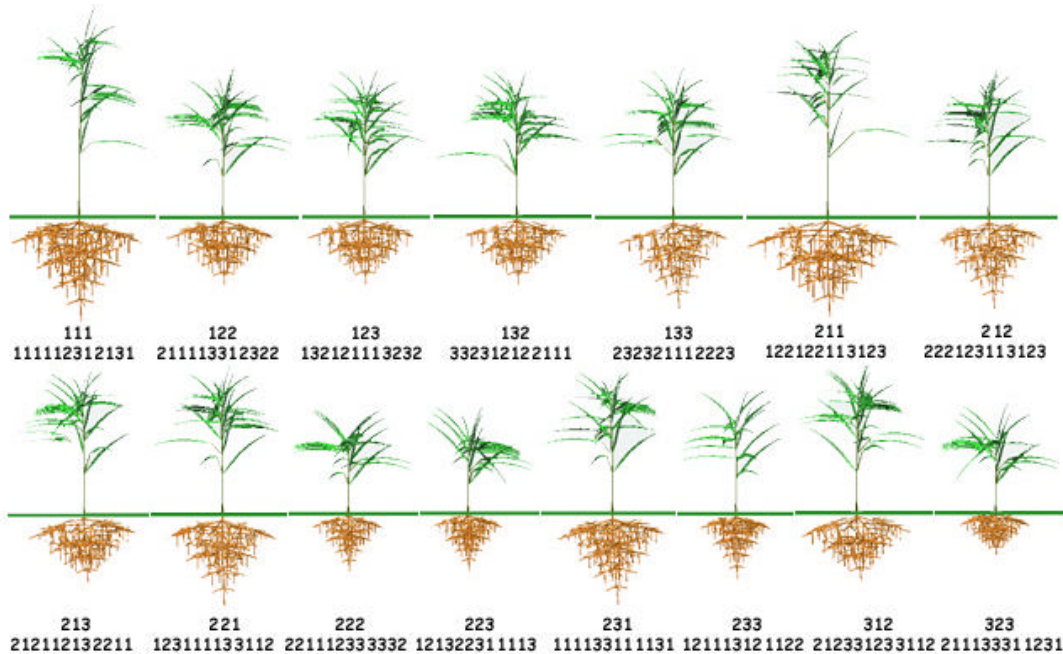


Figure 5. The different plant structure generated by Stochastic L-system.

Fifteen of twenty-seven structures of this plant prototype are given in Figure 5. There are two lines of the number of each plant structure. The first line is the sequence of shoot structure and the root structure is given in the second line. The visualized images of plant growth from time  $t=22$  to  $t=80$  are shown in Figure 6.

The visualization of an example is shown in vegetative state. Cylinders are used to represent internodes and petioles segments. Spheres are used to represent jointed internodes. Triangular polygons are used to represent leaves and flowers.

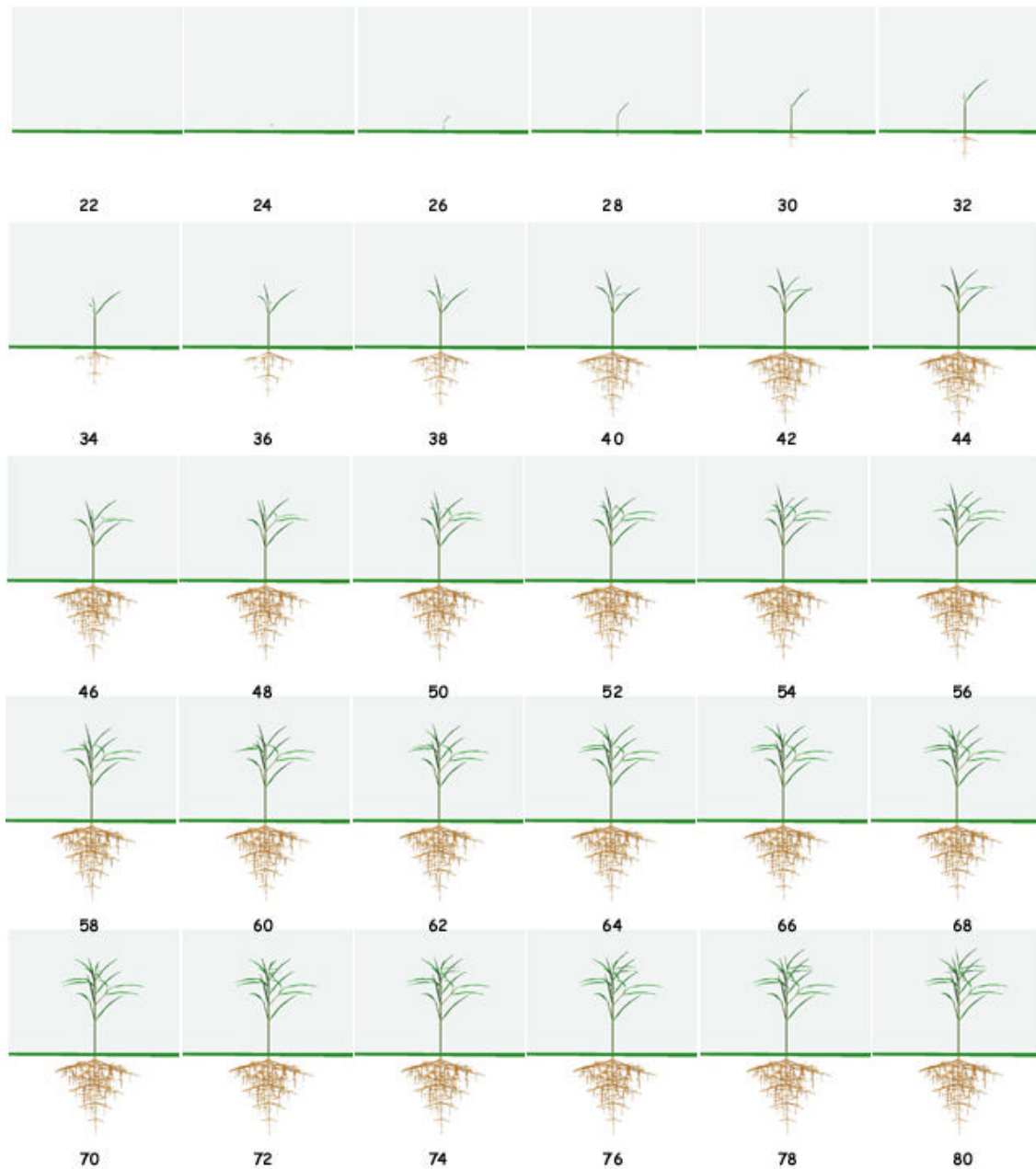


Figure 6. The development of plant growth from time  $t = 22$  to  $t = 80$ .

## 7. Conclusion

A prototype program called **PlantVR** has presented to create the continuous development of plant shoot and root models by parametric functional symbols based on the Bracketed L-system and Stochastic L-system. The optional **Endrule** key word is added to the L-system in order to prevent some symbols that are not defined for plant definition in Table 1. The visualization technique makes the plant looking more realistic and every component can be controlled by the mathematical function. The development of plant growth seems to be smoother and more natural. This prototype can be used to generate a realistic model of the plant. The visualized image and animation of plant development can be visited at this web site.  
<http://www.iwr.uni-heidelberg.de/~schuai>

## References

- [1] H. W. Chen, “L-system Plant Geometry Generator”,  
web site: <http://www.tc.cornell.edu/Visualization/contrib/cs490-94to95/hwchen>
- [2] O. Deussen, P. Hanrahan, B. Linterman, R. Mech, M. Pharr and P. Prusinkiewicz, “Realistic modeling and rendering of plant ecosystems”, *SIGGRAPH'98*, July 1998. pp. 275-286
- [3] C. H. Diaz-Ambrona, A. M. Tarquis and M. I. Minguez, “Faba bean canopy modelling with a parametric open L-system: a comparison with the Monsi and Saeki model”. *Field Crops research*, Vol. 58, No. 1, July 1998. pp. 1-13
- [4] Greenworks, “Home page of the xfrog modelling software”, web site: <http://www.greenworks.de>
- [5] M. S. Hammel and P. Prusinkiewicz, “Visualization of developmental processes by extrusion in space-time”, *Proceedings of Graphics Interface '96*, May 1996. pp. 246-258
- [6] J. Hanan, “Centre for Plant Architecture Informatics”, web site: <http://www.cpai.uq.edu.au>
- [7] B. Horling, “Implementation of a context-sensitive Lindenmayer-system modeler”,  
web site: <http://www2.trincoll.edu/~bhorling/lsystems>
- [8] L. Lapre, “L-systems and Lparser”, website: <http://www.xs4all.nl/~ljlapre>
- [9] T. Lin, “Animation of L-system based 3-D plant Growing in Java”,  
web site: <http://www.cs.umbc.edu/~ebert/693/TLin>
- [10] B. Lintermann and O. Deussen, “A Modelling Method and User Interface for Creating Plants”, *The Eurographics Association*, Vol. 17, No. 1, March 1998. pp. 73-82
- [11] B. Lintermann and O. Deussen, “Interactive Modeling of Plants”, *IEEE Computer Graphics and Applications*, Vol. 19, No. 1, January 1999. pp. 56-65
- [12] R. Mech and P. Prusinkiewicz, “Visual models of plants interacting with their environment”, *Proceedings in Computer Graphics (SIGGRAPH'96)*, August 1996. pp. 397-410
- [13] G. Ochoa, “An Introduction to Lindenmayer Systems”,  
web site: <http://www.cogs.susx.ac.uk/users/gabro/lsys/lsys.html>
- [14] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, New York: Springer-Verlag, 1990.
- [15] P. Prusinkiewicz, M. S. Hammel, J. T. Kajiya and E. Mjolsness, “Animation of plant development”. *Proceedings of Computer graphics SIGGRAPH'93*, August 1993. pp. 351-360
- [16] P. Prusinkiewicz, W. Remphrey, C. Davidson and M. Hammel, “Modeling the architecture of expanding *Fraxinus pennsylvanica* shoots using L-systems”, *Canadian Journal of Botany*, Vol. 72, 1994. pp. 701-714
- [17] P. Prusinkiewicz, M. James and R. Mech, “Synthetic topiary”, *Proceedings of SIGGRAPH' 94*, July 1994. pp. 351-358
- [18] S. Chuai-Aree, S. Siripant and C. Lursinsap, “Animation plant growth in L-system by parametric functional symbols”, *Proceeding of International Conference on Intelligent Technology 2000*, December 2000. pp. 135-143
- [19] S. Chuai-Aree, An algorithm for simulation and visualization of plant shoots growth, Master Thesis, Chulalongkorn University, 2000.