Animating Plant Growth in L-System By Parametric Functional Symbols

Somporn Chuai-aree, Suchada Siripant, and Chidchanok Lursinsap
Advanced Virtual and Intelligent Computing Center (AVIC)
Department of Mathematics
Chulalongkorn University
Bangkok 10330, Thailand
Email: csomporn@math.sc.chula.ac.th, ssuchada@chula.ac.th, lchidcha@chula.ac.th

Abstract: The iteration of L-system has been used to animate the plant growth but at each time step of development the plant model is not smooth and continuous. This paper proposes an animating plant growth in L-system by parametric functional symbols to the length, size and position of each component of the plant. The developments 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 bracketed L-system.

Key words: L-system, Lindenmayer system, realistic image, continuous simulation, simulation and visualization, plant growth, plant model, parametric functional symbols.

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.

In the previous works, Tong Lin implemented the animation of L-system in a 3-dimensional space plant growing in Java [13]. Przemyslaw Prusinkiewicz extended L-systems [6] in manner suitable for simulation to interaction between a developing plant and its environment. They developed the plant growth animation by iterating the L-system but the animation was not smooth.

This research presents a prototype for creating computer models that capture the development of plants using L-systems and mathematical model incorporating biological data. L-system is used for qualitative model in order to represent the plant topology and development. There are six consecutive steps in this method, namely, (1) defining a qualitative model constructed from observations of plant growth in their life cycle, (2) measurement of key characteristics collected 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 identify any incorrectly estimated functions. This research uses soybean (*Glycine max.*) for our case study.

The rest of the paper is organized into 9 sections. Section 2 summarizes the concept of a general L-systems. Section 3 expresses the plant module design. Section 4 describes the qualitative model of plant structure. Section 5 explains the data collection from actual plant. Section 6 shows the growth function approximation. Section 7 describes the quantitative model. Section 8 shows the visualization and Section 9 discusses model evaluation. The conclusion is given in Section 10.

2. L-systems

Lindenmayer systems (L-systems) were first introduced in 1968 by Aristid Lindenmayer as a mathematical theory of plant development [1]. They have attracted the attention of the computer scientists who investigated them through formal language theory [1]. Smith proposed using L-systems as a tool for creating computer generated images of plants. Specialists in computer graphics, particularly Prusinkiewicz have used the L-systems to produce realistic image of trees, bushes, and flowers, and some of image are well illustrated in The Algorithmic Beauty of Plants [1].

L-systems are formal grammars that characterized by having a single axiom, no terminal symbols and by performing all possible symbol replacements in parallel for each step.

A simple L-system called "turtle"[1] is introduced to illustrate how L-system is used to describe some structures.

The basic idea of turtle interpretation described by Prusinkiewicz and Hanan [1] is given below. A turtle can move in any direction, either forward, backward, rightward, or leftward. Each movement is defined using five 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 to movement in counter-clockwise direction while symbol – denotes the movement in clockwise direction. A branch is defined by symbols [and]. Symbol [denotes the begin of branch and symbol] denotes to 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. α can be either positive or negative. The other unit movement can have their 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 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 degrees and δ to 30 degrees.

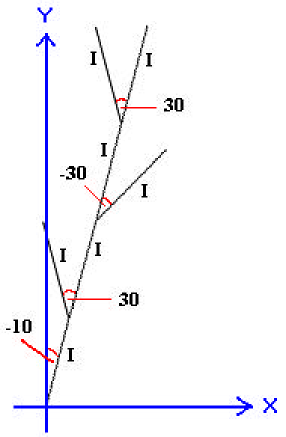

Figure 1: Example of simple 2-dimensional space L-system I[+I]I[-I]I[+I]I

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, 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, 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 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, δ_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 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 /, \, &, ^, +, -, |. The meaning of each symbol is explained in Table 1.

$$R_X(q) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cos(q) & -Sin(q) \\ 0 & Sin(q) & Cos(q) \end{bmatrix}$$

$$R_Y(q) = \begin{bmatrix} Cos(q) & 0 & -Sin(q) \\ 0 & 1 & 0 \\ Sin(q) & 0 & Cos(q) \end{bmatrix}$$

$$R_z(q) = \begin{bmatrix} Cos(q) & Sin(q) & 0 \\ -Sin(q) & Cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 2: Rotation matrices

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 $R_z(\delta_z)$.
-	Turn right by angle δ_z, using rotation matrix $R_z(-\delta_z)$.
&	Pitch down by angle δ_y, using rotation matrix $R_y(\delta_y)$.
^	Pitch up by angle δ_y, using rotation matrix $R_y(-\delta_y)$.
\	Roll left by angle δ_x, using rotation matrix $R_x(\delta_x)$.
/	Roll right by angle δ_x, using rotation matrix $R_x(-\delta_x)$.
|	Turn around, using rotation matrix $R_y(180)$.
[	Push the current state to stack. It is used to set the degree of the plant branch.
]	Pop the current state from stack.

Table 1: Symbols used in plant growth L-system

An example of a simple turtle in a 3-dimensional space is given below.
$(\delta_x = \delta_y = \delta_z = 70,\ \alpha_x = \alpha_y = \alpha_z = 0)$

I[-I]I[+I]I[/I]I[\I]II

We define the symbol *I* for an internode of a plant. An example has ten internodes, six internodes for main stem, and four internodes of petiole in each direction. The visualized image is shown in Figure 3.

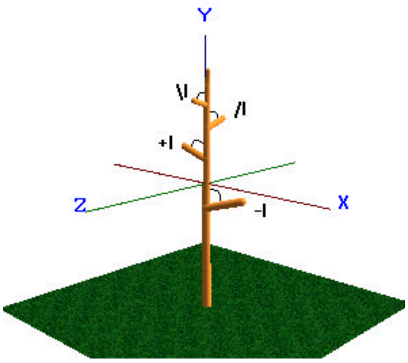

Figure 3: A simple L-system Interpretation

3. Plant module design

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

```
Plant_Name {
        Iterations=N
        Angle=δ
        Diameter=D
        Axiom=ω
        Production 1
        Production 2
        …
        Production n
        Endrule
        Endproduction 1
        Endproduction 2
        …
        Endproduction m
}
```

The meaning of each keyword is given as follows:

Plant_Name

Plant_Name is a name of plant module.

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 a 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=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 (α_x,α_y,α_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=Successor

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

Endrule

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

Endproduction 1 ... Endproduction m

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

Character"{" and "}"

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

4. Qualitative model

The modeling process begins with the specification of the qualitative model. It captures the aspects of a plant which 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 are also defined. In this paper, we obtain the qualitative parameters from a soybean.

The qualitative parameters of the soybean model consists 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

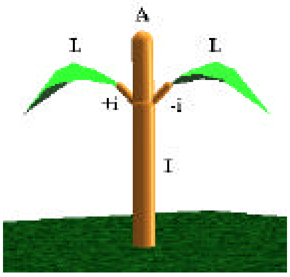

Figure 4: Axiom = I[+iL][-iL]A

initially contained 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.

A = 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 of the left and right petioles are defined as follows.

P = IIII[\pL][/pL][-pL]
B = IIII[\pL][/pL][+pL]

The "Endrule" is defined as follows.

B = IL
P = IL
A = IL

The above *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 in the following format.

```
Soybean {
        Iterations=6
        Angle=45
        Diameter=1.5
        Axiom=I[+iL][-iL]A
        A = I[-P]I[+B]A
        P = IIII[\pL][/pL][-pL]
        B = IIII[\pL][/pL][+pL]
        Endrule
        B = IL
        P = IL
        A = IL
}
```

In the above L-systems code, the number of iteration is six; the initial branch angle is 45 degrees, and the diameter of first internode is 1.5 centimeters. After the sixth iteration, the *Endrule* productions are called to terminate the substitution process.

5. Data collection

The data of each component are collected from an actual soybean plant. They are the internodes length, diameter, leaves length and width, flowers, petioles 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 to the sigmoidal growth function. The data of soybean are collected manually using rulers and a protractor.

6. Growth function

The raw data in Section 5 is approximated as a growth function in Figure 5 using a sigmoidal curve approximation.

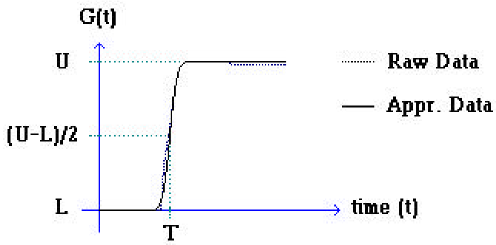

Figure 5: Sigmoidal curve approximation

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

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

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 which we use to control all the components of the plant topology, such as the length of each internode from the first internode to the last internode. The function is

$$Y_i = c(a)^{ni}$$

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*.

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

$$B_i = \boldsymbol{b} n_i + b$$

where B_i is the initial time of component i, $\boldsymbol{b}$ 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 a self-growth function likes the one shown in Figure 3 with either same or different slope m, time T, the maximum U, and the minimum L.

Figure 6 illustrates a graphical image of a soybean drawn from the L-system string defined in the production rules in Section 4. Figure 6(a) shows the string of L-system obtained after the rewriting process. The plant structure in Figure 6(b) is constructed from the L-system string shown in Figure 6(a) and its graphical image of the axiom of the plant is illustrated in Figure 6(g). Figure 6(d) gives the details a part of soybean. Figure 6(c) is the left petiole component. The right petiole is shown in Figure 6(e). In Figure 6(f), the raw data is converted to the growth function corresponding to each symbol.

7. Quantitative model

The quantitative model combines the L-system output string with the approximated growth functions of each component. The plant model can simulate its growth with continuous development in virtual reality form.

8. Visualization

The visualization of a soybean 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 7 shows some selected stages of the development of a soybean shoot controlled by the production rules defined in Section 4. The developments in Figure 7 are started at time t = 20 according to the sigmoidal curve in Figure 5.

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

```
Plant1{
    Iterations=8
    Angle=15
    Diameter=0.8
    Axiom= I[-1][+1][/2][\2][^1][^^1][^\-1][^^\-1]
[^2][^^2]
    1=I[/iL][\iL]1
    2=I[-iL][+iL]2
    endrule
    1=IF
    2=IF
}
```

All plants are generated using eight iterations. Although the same production rules are applied to the plants, it is remarkable that they look like different species. The symbol L and F are linked from our leaf and flower library which are created prior to the generation.

9. Model evaluation

Our system has the capability to interactively adjusting the parameters of the plant model. This allows the designer to verify the production rules and to modify the appearance of the graphical image of the generated plant in a real time mode. In addition, if there are any flaws present in the plant model due to the production rules the designer can edit the rules and recompile the L-system description.

10. Conclusions

A prototype program called *PlantVR* has presented for creating the continuous development of plant models by parametric functional symbols based on the bracketed L-systems using soybean model as a case study. The optional *Endrule* key word is added to the L-systems in order to prevent some symbols that is not defined for plant definition in Table 1. The visualization technique makes the plant look more realistic and every component can be controlled by a mathematical function. This prototype can be used to generate the realistic model of any plant that has life cycle similar to soybean. The visualized image and animation of plant development can be visited at this web site: http://www.avic.sc.chula.ac.th/plantvr/index.htm

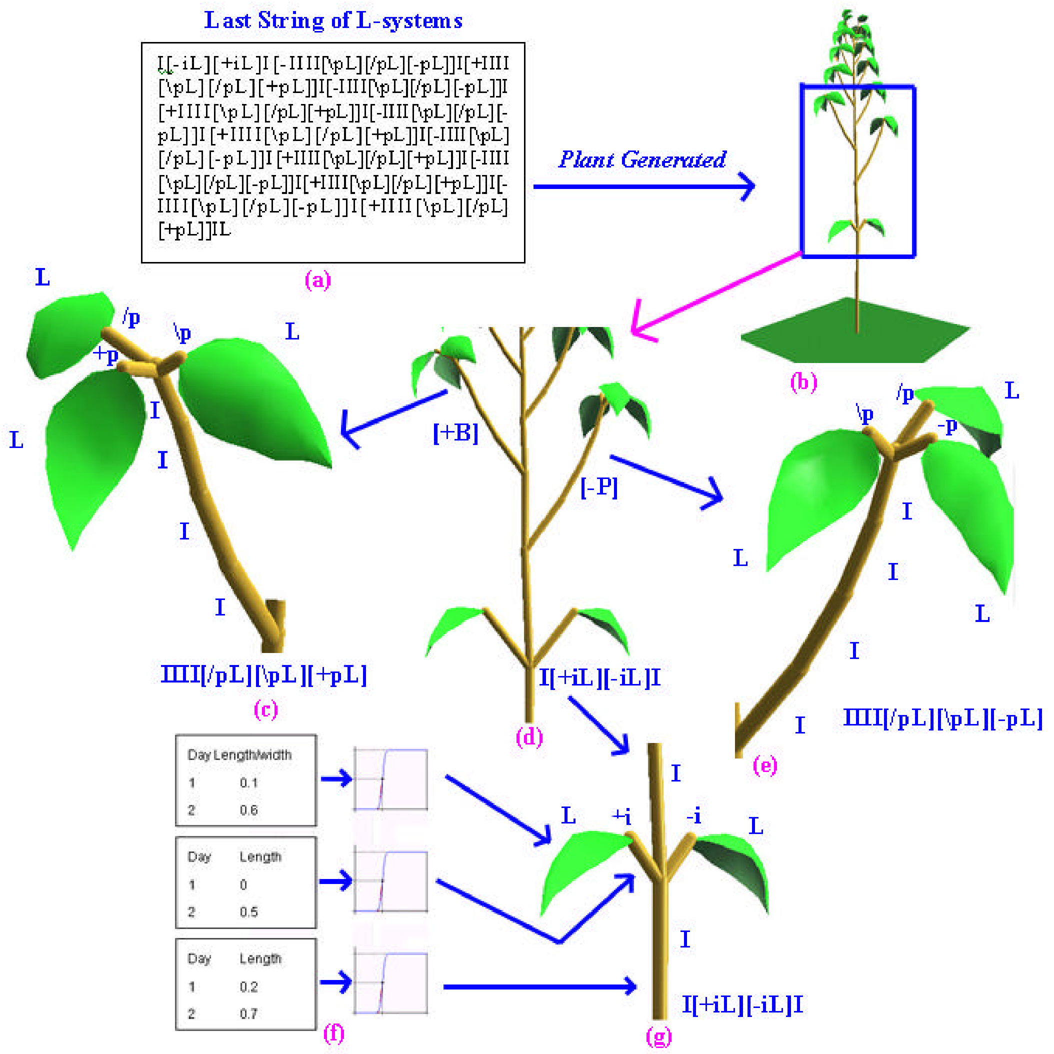

Figure 6: Structure of simulation generated from the production rules of soybean in Section 4

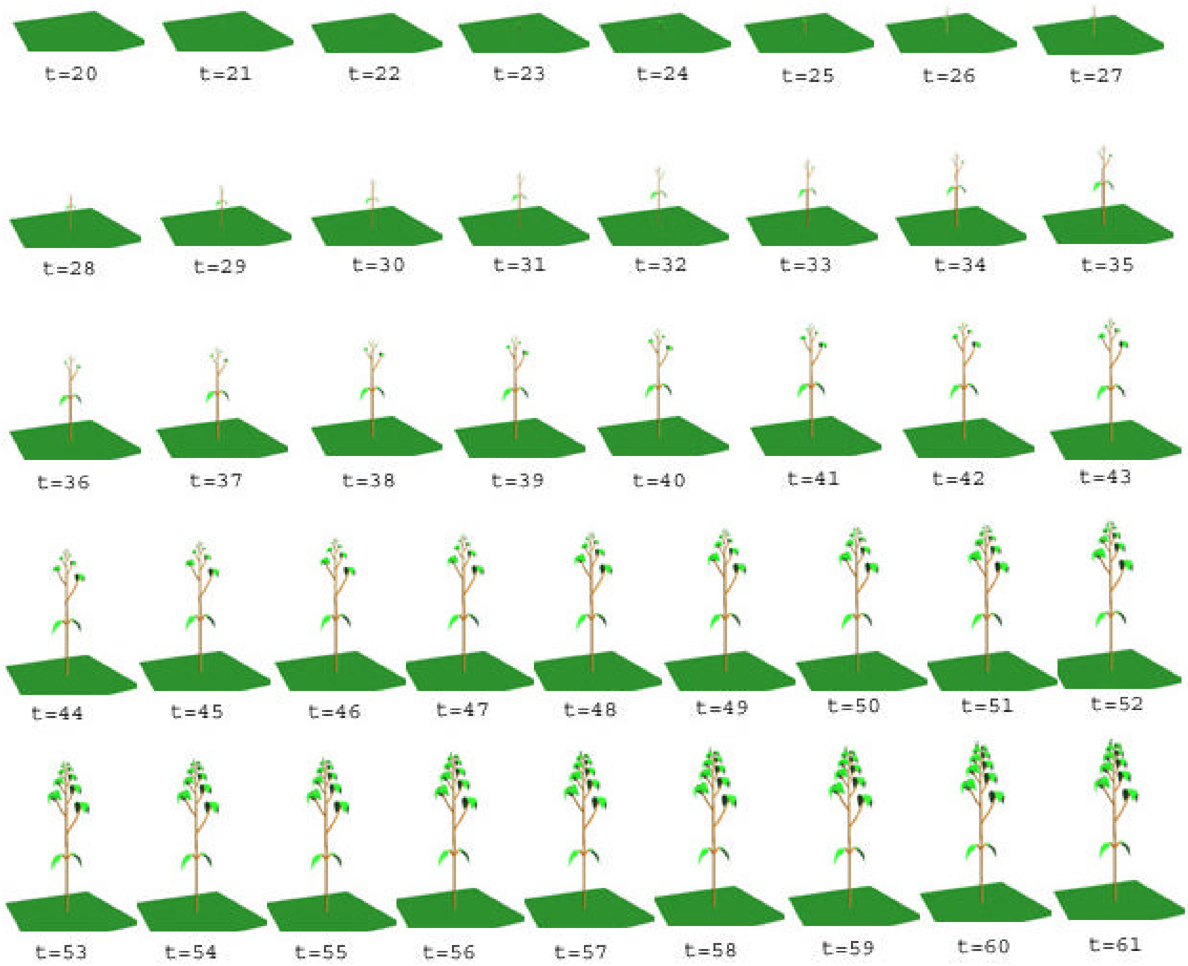

Figure 7: Simulation and visualization of Soybean shoots expansion over 61 days

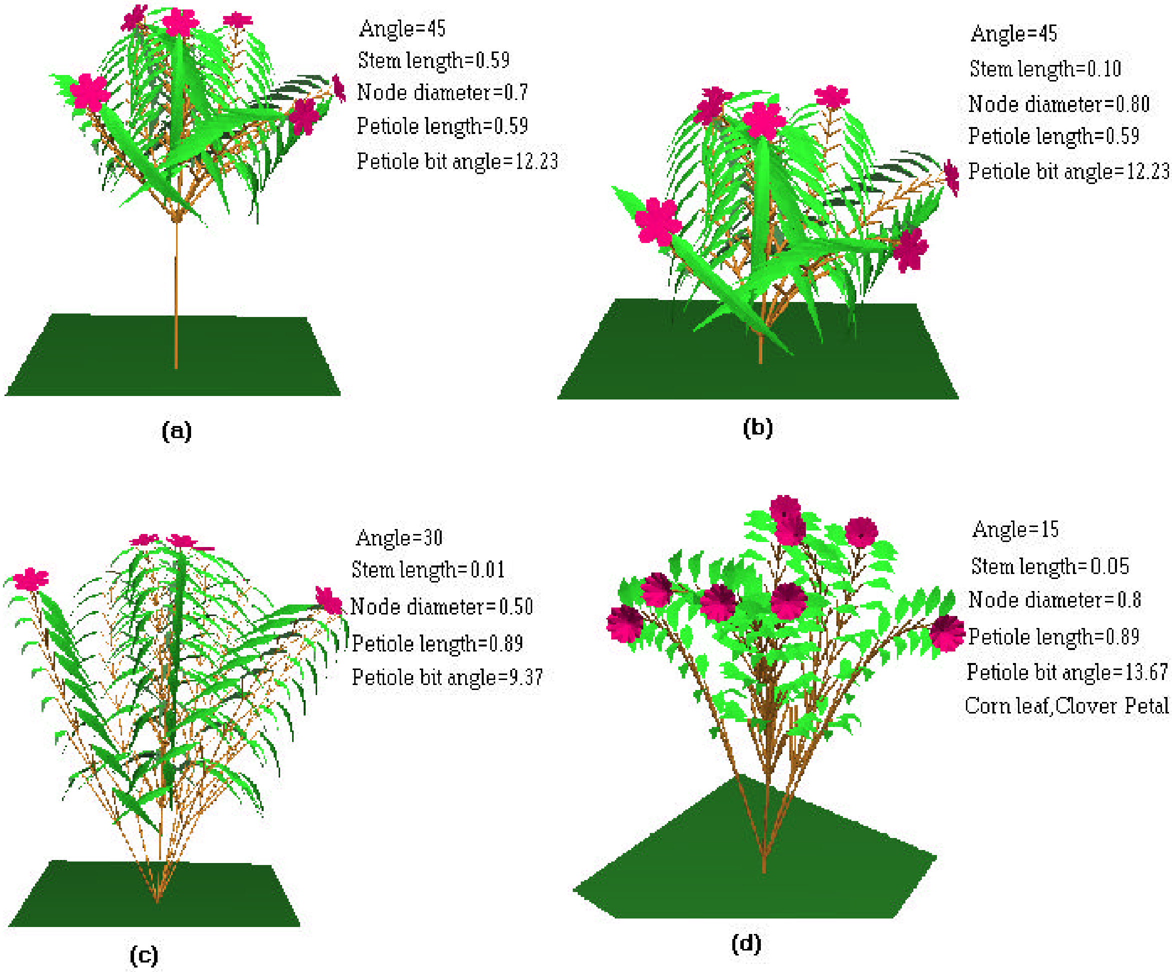

Figure 8: Different parameters of same topology of L-system

Acknowledgements

This work is partially supported by a grant from TJTTP project which gives a collaboration with Prof. Y. Shinagawa of University of Tokyo. We would like to thank Dr. Jim Hanan for his information, suggestions and discussions in this research, Prof. Przemyslaw Prusinkiewicz for kindly giving us the L-studio. We also would like to thank Mr. Salin Phuvipadawat and Dr. Preeda Boon-long for giving information of plant development.

References

[1] Prusinkiewicz, P. and Lindenmayer, A. The Algorithmic Beauty of Plants, New York: *Springer-Verlag*, 1990.

[2] Mech, R., and Prusinkiewicz, P. Visual models of plants interacting with their environment, *in Computer Graphics (SIGGRAPH '96 Proceedings)*, 397-410, August 1996.

[3] Prusinkiewicz, P., Hammel, M.S. and Mjolsness, E. Animation of plant development, *Proceedings of Computer Graphics SIGGRAPH'93*, 351-360, August 1993.

[4] Prusinkiewicz, P., Hammel, M., Hanan, J., and Mech, R. Visual model of plant development, Handbook of formal language. *Springer-Verlag*, 1996.

[5] Hanan, J. Virtual plants---Integrating architectural and physiological plant models, *The Modelling and Simulation Society of Australia, Proceedings of ModSim 95*, 1: 44-50, 1995.

[6] Prusinkiewicz, P., James, M. and Mech, R. Synthetic topiary. *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24-29, 1994), ACM SIGGRAPH*, 351-358, July 1994.

[7] Prusinkiewicz, P. and Kari, L. Subapical bracketed L-systems, *Proceedings of the Fifth International Workshop on Graph Grammars and their Application to Computer Science*, 1073: 550-564, 1994.

[8] Room, P.M. and Hanan, J.S. Virtual cotton: a new tool for research, management and training, *Proceedings of the World Cotton Research Conference*, 14-17, February 1994.

[9] Room, P.M, Hanan, J.S. and Prusinkiewicz, P. Virtual plants: new perspectives for ecologists, pathologists, and agricultrual scientists, *Trends in Plant Science*, 1(1): 33-38, 1996.

[10] Room, P.M, Maillette, L. and Hanan, J. Module and metamer dynamics and virtual plants, *Advances in Ecological Research*, 25: 105-157, 1994.

[11] Lintermann, B. and Deussen, O. A Modelling Method and User Interface for Creating Plants, *The Eurographics Association*, 17(1): 73-82, March 1998.

[12] Lintermann, B. and Deussen, O. Interactive Modeling of Plants, *IEEE Computer Graphics and Applications*, 19(1): 56-65, January 1999.

[13] Lin, T. Animation of L-system based 3-D Plant Growing in Java, *http://www.cs.umbc.edu/~ebert/693/TLin/*

[14] Chuai-aree., S. An algorithm for simulation and visualization of plant shoots growth, *http://www.aviclab.org/plantvr*