ขั้นตอนวิธีสำหรับการจำลองและการสร้างภาพนามธรรมของการเจริญเติบโต ของลำต้นพืช

นายสมพร ช่วยอารีย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคณนา ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2543

ISBN 974-346-469-7

AN ALGORITHM FOR SIMULATION AND VISUALIZATION OF PLANT SHOOTS GROWTH

Mr. Somporn Chuai-aree

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computational Science
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic Year 2000
ISBN 974-346-469-7

Thesis Title AN ALGORITHM FOR SIMULATION AND VISUALIZATION OF PLANT SHOOTS GROWTH

By Mr. Somporn Chuai-aree

Department Mathematics

Thesis Advisor Associate Professor Suchada Siripant

Thesis Co-advisor Professor Chidchanok Lursinsap, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

…..……………………………………… Dean of Faculty of Science

(Associate Professor Wanchai Phothiphichitr, Ph.D.)

THESIS COMMITTEE

…..……………………………………… Chairman

(Assistant Professor Jack Asavanant, Ph.D.)

…..……………………………………… Thesis Advisor

(Associate Professor Suchada Siripant)

…..……………………………………… Thesis Co-advisor

(Professor Chidchanok Lursinsap, Ph.D.)

…..……………………………………… Member

(Assistant Professor Peraphon Sophatsathit, Ph.D.)

สมพร ช่วยอารีย์ : ขั้นตอนวิธีสำหรับการจำลองและการสร้างภาพนามธรรมของการเจริญเติบโตของลำต้นพืช (AN ALGORITHM FOR SIMULATION AND VISUALIZATION OF PLANT SHOOTS GROWTH) อ. ที่ปรึกษา : รศ. สุชาดา ศิริพันธุ์, อ. ที่ปรึกษาร่วม : ศ.ดร. ชิดชนก เหลือสินทรัพย์ ; 205 หน้า. ISBN 974-346-469-7.

การเจริญเติบโตของพืช ณ เวลาต่างๆ ได้มีการทำซ้ำของ Lindenmayer systems หรือ L-systems แต่ภาพเคลื่อนไหวที่ได้จะไม่ราบรื่นและต่อเนื่อง วิทยานิพนธ์นี้ได้เสนอวิธีการสร้างภาพเคลื่อนไหวของการเจริญเติบโตของพืชด้วย L-systems ด้วยวิธี parametric functional symbols เพื่อใช้ในการควบคุม ความยาว ขนาด และตำแหน่งของแต่ละส่วนประกอบของพืชโดยการเพิ่มฟังก์ชันทางคณิตศาสตร์เข้าไป ควบคุมแต่ละสัญลักษณ์ที่แทนส่วนประกอบต่าง ๆ ของพืช ซึ่งจะทำให้การเจริญเติบโตของพืชเป็นไปอย่างราบรื่นและธรรมชาติมากขึ้น โดยที่ตัวแบบ (prototype) นี้สามารถที่จะนำไปสร้างต้นไม้ชนิดอื่นๆ ได้ด้วยหลักการของ bracketed L-system

ภาควิชา คณิตศาสตร์ ลายมือชื่อนิสิต..

สาขาวิชา วิทยาการคณนา ลายมือชื่ออาจารย์ที่ปรึกษา..............................

ปีการศึกษา 2543 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม..........................

SOMPORN CHUAI-AREE : AN ALGORITHM FOR SIMULATION AND VISUALIZATION OF PLANT SHOOTS GROWTH. THESIS ADVISOR: ASSOC. PROF. SUCHADA SIRIPANT AND PROF. CHIDCHANOK LURSINSAP, Ph.D. 205 pp. ISBN 974-346-469-7.

The development of plant growth at each time step has been used the iteration of Lindemayer systems (L-systems), but the animation was not smooth and continuous. This thesis proposed an animating plant growth in L-systems by mean of the parametric functional symbols to control the length, size and position of each component of the plant. As a consequence, the development of plant growth became smoother and more natural as realistic. The resulting prototype can be used to generate a realistic model of any plant based on bracketed L-systems.

Department Mathematics Student's signature……………………………
Field of study Computational Science Advisor's signature……………………………..
Academic year 2000 Co-advisor's signature…………………………..

Acknowledgements

I am greatly indebted to my supervisor, Associate Professor Suchada Siripant and Professor Dr. Chidchanok Lursinsap, for their suggestions, untiring help and guidance of this thesis. Without them my thesis would have never been accomplished.

I also wish to express my special thanks to the thesis committee, Assistant Professor Dr. Jack Asavanant and Assistant Professor Dr. Peraphon Sophatsathit, for their valuable advise, reading and criticizing the manuscript. I would like to thank my superior offices for their permission to take leave, my colleagues at Department of Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University, Pattani Campus.

This work is partially supported by a grant from TJTTP project through a collaboration with Professor Y. Shinagawa of the University of Tokyo, Japan. I would like to thank Dr. Jim Hanan for his information, suggestions and discussions of this research, Professor Przemyslaw Prusinkiewicz for his permission to use the L-studio software, Mr. Salin Phuvipadawt and Dr. Preeda Boon-long for furnishing information of plant growth.

Furthermore, I would like to thank all my past and present instructors for their valuable lecture and instructions. Finally, I would like to express my sincere gratitude to my parents for their love, hearty encouragement and unselfish sacrifice during my study.

I truly believe that all people whom I have not personally mentioned here are aware of my deep appreciation.

Table of Contents

List of Tables

List of Figures

Page

Chapter 1

Introduction

1.1 Problem Identification

In nature, the growth of the plants seems to grow in randomness, and it is very interested in their growing. Time-lapse photography reveals the enormous visual appeal of developing plants, related to the extensive changes in topology and geometry during growth. Consequently, the animation of plant development represents an attractive and challenging problem for computer graphics. We hypothesize that Lindenmayer systems (L-systems) and parametric functional symbols should be efficient for plant development look more realistic.

The L-systems code has been created by John Martin Carroll in 1998. His software supports the deterministic bracketed L-systems and simple L-systems in two-dimensional space.

```
plant2{
  ;from The Algorithmic Beauty of Plants
  #iterations = 6
  dirs = 16
  axiom = ----X
  X = F[+X][-X]FX
  F = FF
}
```

In above L-systems code, The iteration is six, the angle is 16, the axiom is *X*. The letter *X* is replaced by string *F[+X][-X]FX* to the first production, then the string *FF* will replace a letter *F* where a letter *F* represents a internode. The image is created from these parameters. Six frames of plant development from first iteration (n=1) to sixth iteration (n=6) are shown in Figure 1.1.

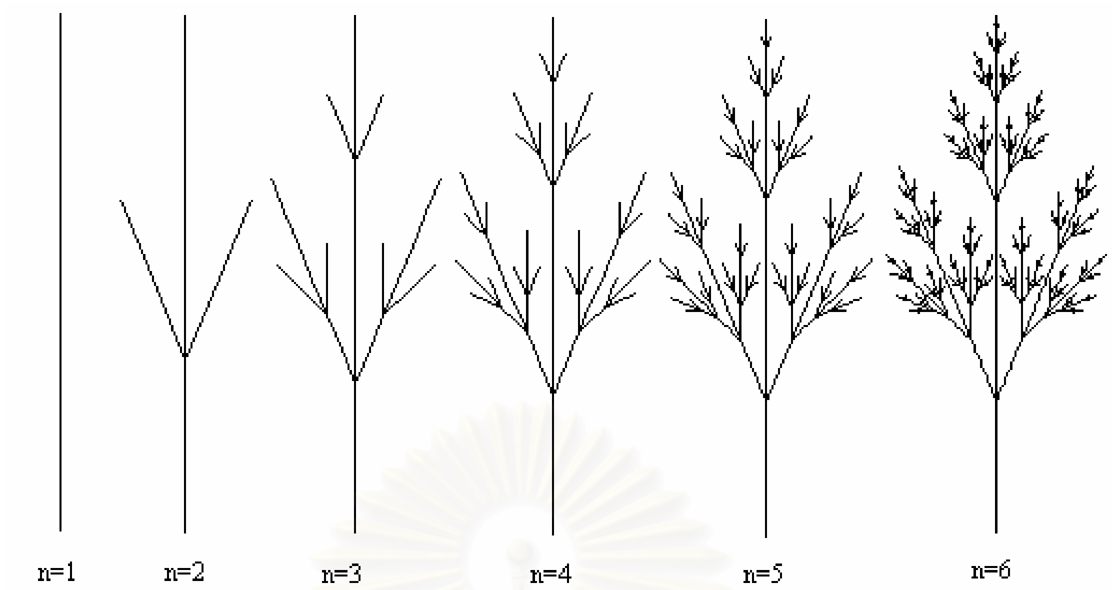


Figure 1.1: The development example of L-systems.

If we animate these six frames of plant development, the animation will not smooth and continuous, because the developments of some internodes are not shown for continuous frame. In order to solve this problem, the parametric functional symbols are applied in this thesis.

The iteration of L-systems has been used to animate the plant growth but at each time step of development the plant model was not smooth and continuous. This thesis presents a prototype to simulate and visualize the plant growth in L-systems by parametric functional symbols to the length, size and position of each component of plant, it can be seen that the plant model looks more realistic.

The parametric functional symbols are the symbols that added the parameter function to control the graphic form of plant development. It will be described in Chapter 4.

1.2 Objectives of the Research

The main objectives of this study are the following:

1. To develop an algorithm of plant development.
2. To simulate and visualize plant development by implementing in virtual reality form.

1.3 Scope of the Research

In the environment of plant development; light, carbon dioxide, water and soil are needed. The future plan of our project is to study how these factors are important and effective to plant development. This research will develop a prototype of plant development from data that are collected from some experiments which ignore all the factors of plant growing by using soybeans as case study. We measure the structural development of plants as individuals made up of components like apices length, internodes length, leaves width, leaves length, and diameter at different time steps in their life cycle. The mathematical model of soybean development will be simulated in virtual reality form. This prototype can be used to generate the realistic model of any plant based on bracketed L-systems.

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-systems 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) measuring 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) visualizing of the quantitative model, and (6) evaluating model.

1.4 Details Schedule

The details schedule of this thesis are the following:

1. Search and study previous works about plant development and L-systems.
2. Collect data from soybean experiments.
3. Analyze the raw data to approximate soybean growth using growth function.
4. Study computer graphics, delphi programming, and OpenGL graphics library.
5. Write program to visualize the plant growing.

6. Experiment another plant to test the prototype and adjust the prototype for any plants.
7. Conclusion.

1.5 Expected Outcome

The usefulness of this work is to obtain an algorithm to generate and simulate the plant development which can be used to study how plant is develop.

This thesis is organized into six chapters. Chapter 2 reviews the literature. Chapter 3 is theoretical background about the classes of L-systems. The plant module and experimental design are discussed and illustrated in Chapter 4. The visualization procedure and results are shown in Chapter 5. Some final thoughts are summarized in Chapter 6.

Chapter 2

Literature Review

This chapter describes the survey of paper that used the L-systems to applied for plant model and plant development. There are a lot of researches that use L-systems to simulate and visualize the plant model and development. This chapter consists of two sections, a first one reviews about the research that related to L-systems, and the last one describes the research about plant model.

2.1 Review of Literature Related to L-systems

In 1968, Aristid Lindenmayer introduced L-systems, which provided a mathematical formalism for parallel grammars well adapted to the modeling of growth phenomena [27]. In 1984, Alvy Ray Smith, a computer graphics researcher showed how L-systems could be used to synthesize realistic images. He also pointed out the relationship between the concept of Fractals and L-systems [35]. L-systems used to generate plants with or without inflorescence, cell growth and geometric patterns such as Indian kolams or mathematical 'monsters' such as the Von Koch or Hilbert curves. Many geometric patterns and tilings can be generated using L-systems. The problem of describing patterns and tilings using L-systems is largely unexplored in [27].

In 1988, Friedell and Schulmann developed a prototype for the automatic generation of architectural scenes. Which would allow the infinite range of forms that can be generated using L-systems formalism to be explored.

- One set of rules (in the simplest example, only one rule) describes the pattern to be repeated.
- Another rule describes the expansion of the pattern through translations

- And a third rule describes the rotational symmetries. Using the basic structure, all symmetry groups can be described easily using the same grammar skeleton.

The rule-based language of “stochastic sensitive growth grammars” as an extension of parametric L-systems [27] was developed to describe algorithmically the changement of the morphology of forest trees in time, taking endogenous and exogenous factors into account, and to create systematically three-dimensional simulations of tree crowns. At different tree species, mainly at spruce, morphological measurements were carried out to get a basis for the design and parameterization of such rule systems.

The software GROGRA (Growth Grammar Interpreter) creates time series of three-dimensional crown structures [14,15] from the rules; the basic elements of these structures (annual shoots) can additionally bean non-geometrical attributes. Furthermore, GROGRA contains several analysis tools and data interfaces.

The generated architectures serve as an “ecomorphological basis model” for different process-oriented simulation models. There is already realized a model of tree-internal water flow (HYDRA) [15], based on the artificial tree structures.

Lin implemented the animation of L-systems based on three-dimensional plant growing in Java [17]. His animation used a number of iteration to animate the plant development. His animation was not smooth.

Prusinkiewicz, James, and Mech extended Lindenmayer systems [29] in manner suitable for simulation the interaction between a developing plant and its environment. The formalism was illustrated by modeling the response of trees to pruning, which yields synthetic images of sculptured plants found in topiary gardens.

Hammel and Prusinkeiwicz extended the notation of L-systems with turtle interpretation [8] to facilitate the construction of such objects. The extension was based on the interpretation of the entire derivation graph generated by an L-systems, as opposed to the interpretation of individual words. The illustrated the proposed method by applying it to visualize the development of compound leaves, a sea shell win a pigmentation pattern, and a filamentous bacteria.

Prusinkiewicz extended it further to language-restricted iterated function systems (LRIFS’s) [24]. They generalized the original definition of IFS’s by providing mean for restricting the sequences of applicable transformations. The resulting attractors include sets that cannot be generated using ordinary IFS’s. Their research was expressed using the terminology of formal languages and finite automata.

Prusinkiewicz, Hammel, and Mjolsness introduced a combined discrete/ continuous model of plant development that integrates L-system-style productions and differential equations [28]. The model was suitable for animating simulated developmental processes in a manner resembling time-lapse photography. The proposed techniques were illustrated using several developmental models, including the flowering plants.

Prusinkiewicz and Kari expressed the development of modular branching structures [26] that satisfy three assumptions: (a) subapical branching, meaning that new branches can be created only near the apices of the existing branches, (b) finite number of module types and states, and (c) absence of the interactions between coexisting components of the growing structure. These assumption were captured in the notion of subapical bracketed deterministic L-systems without interactions (sBOL-systems). They presented the biological rationale for sBOL-systems and proved that it is decidable whether a given BOL-system was subapical or not.

Hemmel, Prusinkiewicz, Remphrey, and Davidson presented a methodology for creating models that capture the development of plant using the formalism of L-systems and incorporating biological data using *Fraxinus pennsylvanical* shoots based on L-systems [30].

Hammel, Prusinkiewicz, and Wyvill proposed a method for modelling compound leaves in plants [9]. The layout of leaf lobes is captured by a branching skeleton generated using an L-systems. The leaf margin is then traced around the skeleton. Their work focused on the specification and tracing of the margin, and including references to the techniques described in the literature for performing the other tasks. The margin is defined as an implicit contour.

2.2 Review of Literature Related to Plant Model

Lintermann and Deussen presented a modelling method and graphical user interface for the creation of natural branching structures such as plants [19,20]. Structural and geometric information is encapsulated in objects that are combined to form a description of the model. The model was represented graphically as a structure graph and could be edited interactively. Global and partial constraint techniques were integrated on the basis of tropism, free-form deformations and pruning operations to allow the modelling of specific shapes.

Deussen, O. developed a system built around a pipeline of tools [3]. The terrain was designed using an interactive graphical editor. Plant distributing was determined by hand, by ecosystem, or by a combination of both techniques.

These two topics of literature review have stimulated the idea for the researcher to improve the previous works. The improvement of the previous works will be discussed later.

Chapter 3

Lindenmayer Systems

This chapter describes the classes of Lindenmayer Systems. Lindenmayer systems (L-systems) were conceived as a mathematical theory of plant development. Originally, they did not include enough detail to allow for comprehensive modeling of higher plants. The emphasis was on plant topology, that is, the neighborhood relations between cells or larger plant modules. Their geometric aspects were beyond the scope of the theory. Subsequently, several geometric interpretations of L-systems were proposed with a view to turning them into a versatile tool for plant modeling.

3.1 Rewriting systems

The main concept of L-systems is rewriting. The rewriting is a technique for defining complex objects by successively replacing parts of a simple initial object using a set of *rewriting rules or productions* [27]. The classic example of graphical object defined in terms of rewriting rules is the *snowflake curve* in Figure 3.1, proposed in 1905 by von Koch [27]. Mandelbrot restates this construction as follow:

One begins with *two shapes*, an *initiator* and a *generator*. The latter is an oriented broken line make up of N equal sides of length r. Thus each stage of the construction begins with a broken line and consists in replacing each straight interval with a copy of the generator, reduced and displaced so as to have the same end points as those of the interval being replaced.

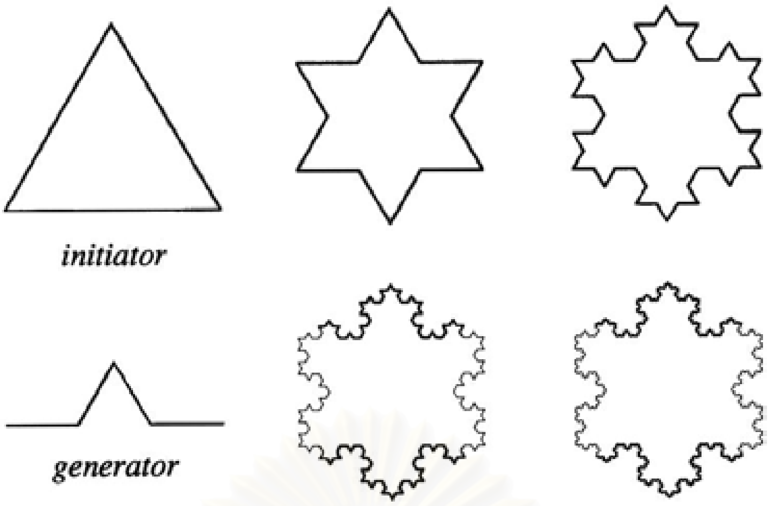


Figure 3.1: Construction of the snowflake curve.

In 1968 a biologist, Aristid Lindenmayer, introduced a new type of string-rewriting mechanism, subsequently termed L-systems [27]. The essential difference between Chomsky grammars and L-systems lies in the method of applying productions. In Chomsky grammars productions are applied sequentially, whereas in L-systems they are applied in parallel and simultaneously replace all letters in a given word. This difference reflects the biological motivation of L-systems. Productions are intended to capture cell divisions in multi-cellular organisms, where many divisions may occur at the same time. Parallel production application has an essential impact on the formal properties of rewriting systems.

3.2 Deterministic and Context-Free L-systems

This section presents deterministic and context-free L-systems (DOL-systems) which are the simplest class of L-systems. The discussion starts with an example that introduces the main concept in intuitive terms.

Consider strings built of two symbols *a* and *b*, which many occur many times in a string. Each symbol is associated with a rewriting rule. The rule $a \rightarrow ab$ means that the letter *a* is to be replaced by the string *ab*, and the rule $b \rightarrow a$ means that the letter *b* is to be replaced by *a*. The rewriting process starts from a distinguished string called the axiom. Assuming that it consists of a single letter *b*. In the first derivation step (the first step of rewriting, n=1), the axiom *b* is replaced by a using production

$b \rightarrow a$. In the second step (n=2) *a* is replaced by *ab* using production $a \rightarrow ab$. The word *ab* consists of two letters, both of which are *simultaneously* replaced in the next derivation step. Thus, *a* is replaced by *ab, b* is replaced by *a,* and the string *aba* results. In a similar way, the string *aba* yields *abaab* which in turn yields *abaababa,* then *abaababaabaab*, and so on as illustrated in Figure 3.2.

$n = 5$
$\omega : b$
$p_1 : a \rightarrow ab$
$p_2 : b \rightarrow a$

Figure 3.2: Example of a derivation in a DOL-system.

Formal definitions describing DOL-systems and their operation are given below. For more details see [27]. Let *V* denote an alphabet, V^* the set of all words over *V*. A *string OL-system* is an ordered triplet $G = <V,\omega ,P>$ where *V* is the *alphabet* of the system, $\omega \in V^+$ is a nonempty word called the *axiom* and $P \subset V \times V^*$ is a finite *set of productions*. A production $(a,\chi) \in P$ is written as $a \rightarrow \chi$. The letter *a* and the word χ are called the *predecessor* and the *successor* of this productions, respectively. It is assumed that for any letter $a \in V$, there is at least one word $\chi \in V^*$ such that $a \rightarrow \chi$. If no production is explicitly specified for a given predecessor $a \in V$, the *identity production* $a \rightarrow a$ is assumed to belong to the set of productions *P*. An OL-system is *deterministic* (noted *DOL-system*) if and only if for each $a \in V$ there is exactly one $\chi \in V^*$ such that $a \rightarrow \chi$.

Let $\mu = a_1 \ldots a_m$ be an arbitrary word over *V*. The word $\nu = \chi_1 \ldots \chi_m \; \varepsilon \; V^*$ is *directly generated* by μ, noted $\mu \rightarrow \nu$, if and only if $a_i \rightarrow \chi_1$ for all $i = 1,\ldots,m$.

A word ν is generated by G in a derivation of *length n* if there exists a *developmental sequence* of words $\mu_0, \mu_1, \ldots, \mu_n$ such that $\mu_0 = \omega$, $\mu_n = \nu$ and $\mu_0 \rightarrow \mu_1 \rightarrow \ldots \rightarrow \mu_n$.

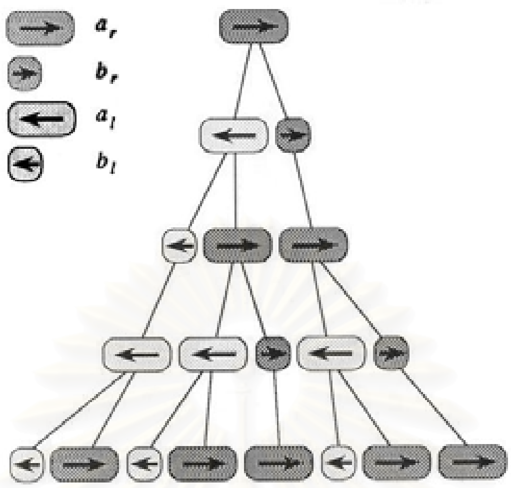


Figure 3.3: Development of a filament (*Anabaena catenula*) simulated using a DOL-system.

The following example provides another illustration of the operation of DOL-systems. The formalism is used to simulate the development of a fragment of a multi-cellular filament such as that found in the blue-green bacteria *Anabaena catenula* and various algae [27]. The symbols ***a*** and ***b*** represent cytological states of the cells (their size and readiness to divide). The subscripts ***l*** and ***r*** indicate cell polarity, specifying the positions in which daughter cells of type ***a*** and ***b*** will be produced. The development is described by the following L-system:

$n = 4$
$\omega : a_r$
$p_1 : a_r \rightarrow a_l b_r$
$p_2 : a_l \rightarrow b_l a_r$
$p_3 : b_r \rightarrow a_r$
$p_4 : b_l \rightarrow a_l$

Starting from a single cell $\boldsymbol{a_r}$ (the axiom), the following sequence of words is generated:

a_r

$a_l\ b_r$

$b_l\ a_r\ a_r$

$a_l\ a_l\ b_r\ a_l\ b_r$

$b_l\ a_r\ b_l\ a_r\ a_r\ b_l\ a_r\ a_r$

. . .

Under a microscope, the filaments appear as a sequence of cylinders of various lengths, with *a*-type cells longer than *b*-type cells. The corresponding schematic image of filament development is shown in Figure 3.3. Note that due to the discrete nature of L-systems, the continuous growth of cells between subdivisions is not captured by this model.

3.3 Turtle interpretation of strings

In order to model the higher plants, a more sophisticated graphical interpretation of L-systems in needed. The first results in this direction were published in 1974 by Frijters and Lindenmayer, and Hogeweg and Hesper. In both cases, L-systems were used primarily to determine the branching topology of the modeled plants. The geometric aspects, such as the lengths of line segments and the angle values, were added in a post-processing phase. The results of Hogeweg and Hesper were subsequently extended by Smith, who demonstrated the potential of L-systems for realistic image synthesis.

The basic idea of turtle interpretation is given below. A *state* of the *turtle* is defined as a triplet (x,y,α), where the Cartesian coordinates $(x$,y) represent the turtle's *position*, and the angle α, call the *heading*, is interpreted as the direction in which the turtle is facing. Given the *step size d* and the *angle increment* δ, the turtle can respond to commands represented by the following symbols in Table 3.1.

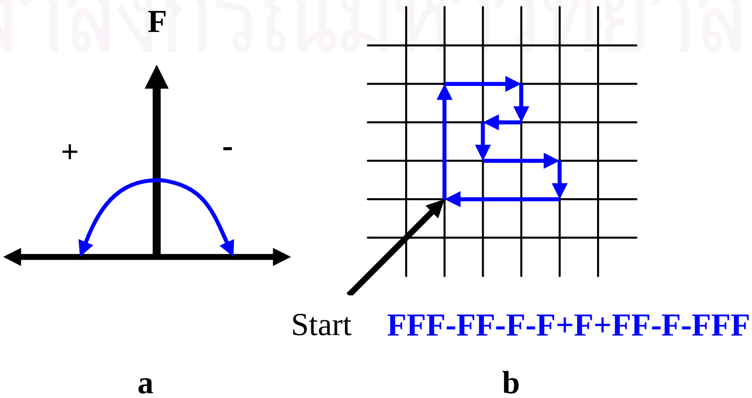


Figure 3.4: Turtle Interpretation of String.

Table 3.1: The two-dimensional turtle interpretation.

Symbols	Meaning
F	Move forward a step of length *d*. The state of the turtle changes to (x', y', α), where $x' = x + d\cos\alpha$ and $y' = y + d\sin\alpha$. A line segment between points (x,y) and (x',y') is drawn.
f	Move forward a step of length *d* without drawing a line.
+	Turn left by angle δ. The next state of the turtle is $(x, y, \alpha+\delta)$. The position orientation of angles is counter-clockwise.
-	Turn right by angle δ. The next state of the turtle is $(x, y, \alpha-\delta)$. The position orientation of angles is clockwise.

Given a string ν, the initial state of the turtle (x_0, y_0, α_0) and fixed parameters *d* and δ, the *turtle interpretation* of ν is the figure (set of lines) drawn by the turtle in response to the string ν in Figure 3.4. Specifically, this method can be applied to interpret strings which are generated by L-systems. For example, Figure 3.5 presents four approximations of the *quadratic Koch island*. These figures were obtained by interpreting strings generated by the following L-system:

$$\omega : F\text{-}F\text{-}F\text{-}F$$

$$p : F \rightarrow F\text{-}F+F+FF\text{-}F\text{-}F+F$$

The images correspond to the strings obtained in derivations of length 0 to 3. The angle increment δ is equal to 90. The step length *d* is decreased four times between subsequent images, making the distance between the endpoints of the successor polygon equal to the length of the predecessor segment.

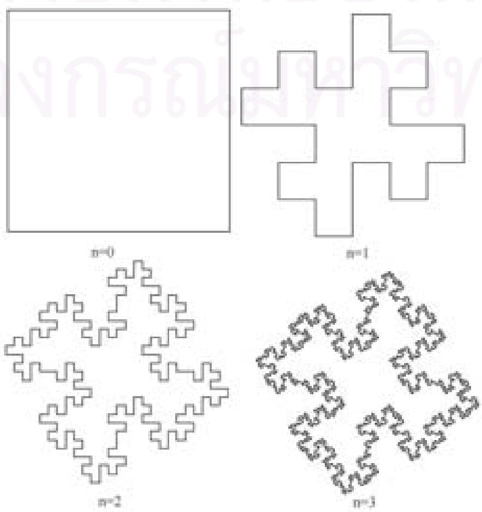


Figure 3.5: Generating a quadratic Koch island.

3.4 Modeling in three dimensions

Turtle interpretation of L-systems can be extended to three dimensions following the ideas of Abelson and diSeassa [27]. The key concept is to represent the current *orientation* of the turtle in space by three vectors

$$\overrightarrow{X}, \overrightarrow{Y}, \overrightarrow{Z},$$

indicating the direction in X-axis, Y-axis, Z-axis, respectively. These vectors have orthorgonal unit vector that satisfy the equation

$$\vec{X} \times \vec{Y} = \vec{Z}$$

Rotations of the turtle are expressed by the equation

[X' Y' Z'] = [X Y Z] R

where R is a 3x3 rotation matrix. The rotations by angle θ about vectors X, Y, Z are represented by the following matrices:

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

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

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

The following symbols control turtle orientation in three-dimensional space as Figure 3.6:

Table 3.2: The symbol of three-dimensional turtle interpretation.

<table>
<tr><th>Symbols</th><th>Meaning</th></tr>
<tr><td>+</td><td>Roll counter-clockwise to positive Z-axis by angle δ_z, using rotation matrix $R_z(\delta_z)$.</td></tr>
<tr><td>-</td><td>Roll clockwise to positive Z-axis by angle δ_z, using rotation matrix $R_z(-\delta_z)$.</td></tr>
<tr><td>&</td><td>Roll counter-clockwise to positive Y-axis by angle δ_y, using rotation matrix $R_y(\delta_y)$.</td></tr>
<tr><td>^</td><td>Roll clockwise to positive Y-axis by angle δ_y, using rotation matrix $R_y(-\delta_y)$.</td></tr>
<tr><td>\</td><td>Roll counter-clockwise to positive X-axis by angle δ_x, using rotation matrix $R_x(\delta_x)$.</td></tr>
<tr><td>/</td><td>Roll clockwise to positive X-axis by angle δ_x, using rotation matrix $R_x(-\delta_x)$.</td></tr>
<tr><td>|</td><td>Turn around, using rotation matrix $R_y(180)$.</td></tr>
</table>

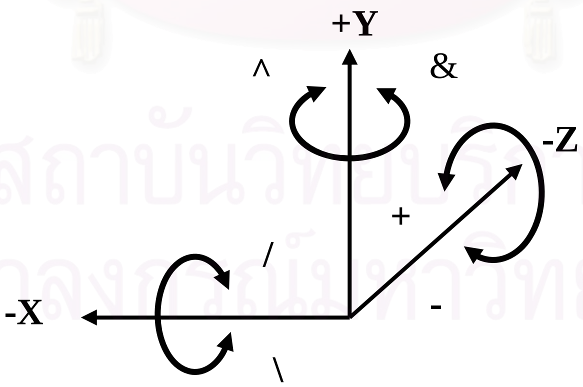


Figure 3.6: Controlling the turtle in three dimensions.

3.5 Bracketed OL-systems

The definition of tree L-systems does not specify the data structure for representing axial trees. One possibility is to use a list representation with a tree topology. Alternatively, axial tree can be represented using *strings with brackets* [27]. A similar distinction can be observed in Koch constructions, which can be implemented either by rewriting edges and polygons or their string representations. An extension of turtle interpretation to strings with brackets and the operation of bracketed L-systems [27] are described below.

Two new symbols are introduced to delimit a branch. They are interpreted by the turtle as follows:

Table 3.3: The bracketed symbols.

Symbols	Meaning
[	Push the current state of the turtle onto a pushdown stack. The information saved on the stack contains the turtle's position and orientation, and possibly other attributes such as the color and width of lines being drawn.
]	Pop a state from the stack and make it the current state of the turtle. No line is drawn, although in general the position of the turtle changes.

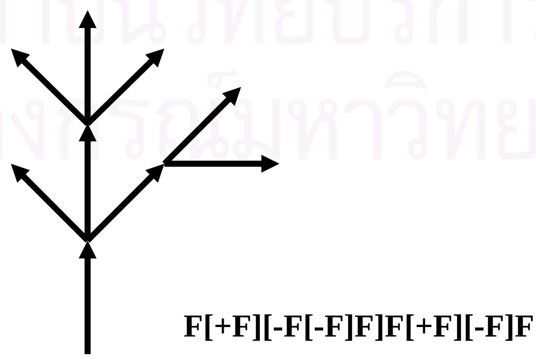


Figure 3.7: Bracketed string representation of an axial tree.

An example of an axial tree and its string representation are shown in Figure 3.7. Derivations in bracketed OL-systems proceed as in OL-systems with out brackets. The brackets replace themselves. Examples of two-dimensional branching structures generated by bracketed OL-systems are shown in Figure 3.8.

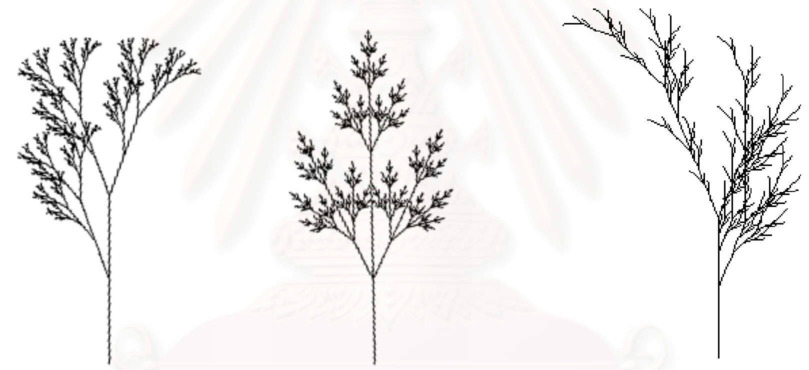
Figure 3.8 illustrates some examples of a three-dimensional bush-like structure generated by a bracketed L-system [27]. Production p_1 creates three new branches from an apex of the old branch. A branch consists of an edge *F* forming the initial internode, a leaf *L* and an apex *A* (which will subsequently create three new branches). Productions p_2 and p_3 specify internode growth. In subsequent derivation steps, the internode gets longer and acquires new leaves. This violates a biological rule of *subapical growth*, but produces an acceptable visual effect in a still picture. Production p_4 specifies the leaf as a filled polygon with six edges. Its boundary is formed from the edges *f* enclosed between the braces { and }. The symbols ! and ' are used to decrement the diameter of segments and increment the current index to the color table, respectively.

Another example of a three-dimensional plat is shown in Figure 3.9. The L-systems can be described and analyzed in a similar manner.

(A) n=5,δ=25.7
F
F→ F[+F]F[-F]+F

(B) n=5,δ=20
F
F→ F[+F]F[-F][F]

(C) n=4,δ=22.5
F
F→ FF-[-F+F+F]+[+F-F-F]



(D) n=7,δ=20
X
X→ F[+X]F[-X]+X
F→ FF

(E) n=7,δ=25.7
X
X→ F[+X][-X]FX
F→ FF

(F) n=5,δ=22.5
X
X→ F-[[X]+X]+F[+FX]-X
F→ FF

Figure 3.8: Examples of plant-like structures generated by bracketed OL-systems.

n=7,δ=22.7
ω=A
p1 : A → [&FL!A]/////’[&FL!A]///////’[&FL!A]
p2 : F → S/////F
p3 : S → FL
p4 : L → [‘’’^^{-f+f+f-|-f+f+f}]

Figure 3.9: A three-dimensional bush-like structure.

3.6 Stochastic L-systems

All plants generated by the same deterministic L-system are identical. An attempt to combine them in the same picture would produce a striking, artificial regularity. In order to prevent this effect, it is necessary to introduce specimen-to-specimen variations that will preserve the general aspects of a plant but will modify its details.

Variation can be achieved by randomizing the turtle interpretation, the L-system, or both. Randomization of the interpretation alone has a limited effect. While the geometric aspects of a plant such as the stem lengths and branching angles are modified, the underlying topology remains unchanged. In contrast, stochastic application of productions may affect both the topology and the geometry of the plant.

A *stochastic OL-system* is an ordered quadruplet $G_\pi = \langle V, \omega, P, \pi \rangle$. The alphabet *V*, the axiom ω and the set of productions *P* are defined as in an OL-system. Function

$\pi : P \rightarrow (0,1]$, called the *probability distribution*, maps the set of productions into the set of *production probabilities*. It is assumed that for any letter $a \varepsilon V$, the sum of probabilities of all productions with the predecessor *a* is equal to 1.

We will call the derivation $\mu \rightarrow \nu$ a *stochastic derivation* in G_π if for each *occurrence* of the letter *a* in the word μ the probability of applying production *p* with predecessor *a* is equal to $\pi(p)$. Thus, different productions with the same predecessor can be applied to various occurrences of the same letter in one derivation step.

A simple example of a stochastic L-system is given below.

$$
\begin{array}{ll}
\omega : & F \\
p_1 : & F \stackrel{.33}{\rightarrow} F[+F]F[-F]F \\
p_2 : & F \stackrel{.33}{\rightarrow} F[+F]F \\
p_3 : & F \stackrel{.34}{\rightarrow} F[-F]F
\end{array}
$$

The production probabilities are listed above the derivation symbol →. Each production can be selected with approximately the same probability of 1/3. Examples of branching structures generated by this L-system with derivations of length 5 are shown in Figure 3.9. Note that these structures look like different specimens of the same plant species.

3.7 Edge and Node rewriting

Random modification of productions gives little insight into the relationship between L-systems and the figures they generate. However, we often wish to construct an L-systems which captures a given structure or sequence of structures representing a developmental process. This is called the *inference problem* in the theory of L-systems. Although some algorithms for solving it were reported in the literature [27], the are still too limited to be of practical value in the modeling of higher plants. Consequently, the methods introduced below are more intuitive in nature. They exploit two modes of operation for L-systems with turtle interpretation, called *edge rewriting* and *node rewriting* using terminology borrowed from graph grammars [27]. In the case of edge rewriting, productions substitute figures for polygon edges, while node rewriting, productions operate on polygon vertices. Both approaches rely on capturing the recursive structure of figures and relating it to a tiling of a plane. Although the concepts are illustrated using abstract curves, they apply to branching structures found in plants as well.

Edge rewriting identifies edges as specific types of edges, which the turtle does not interpret, but the different types of edges affect rewriting rules.

Example:

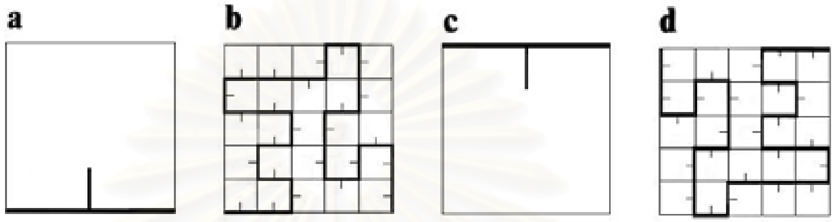


Figure 3.10: Construction of the E-curve on the square grid.
a) F_l **b)** $F_l \rightarrow F_lF_l + F_r + F_r - F_l - F_l + F_r + F_rF_l - F_r - F_lF_lF_r + F_l - F_r - F_lF_l - F_r + F_lF_r + F_r + F_l - F_l - F_rF_r +$
c) F_r **d)** $F_r \rightarrow - F_lF_l + F_r + F_r - F_l - F_lF_r - F_l + F_rF_r + F_l + F_r - F_lF_rF_r + F_l + F_rF_l - F_l - F_r + F_r + F_l - F_l - F_rF_r$

Node rewriting substitutes new polygons for nodes of the predecessor curve.
Example:

$$L_{n+1} = +R_nF - L_nFL_n - FR_n+$$
$$R_{n+1} = -L_nF + R_nFR_n + FL_n-$$

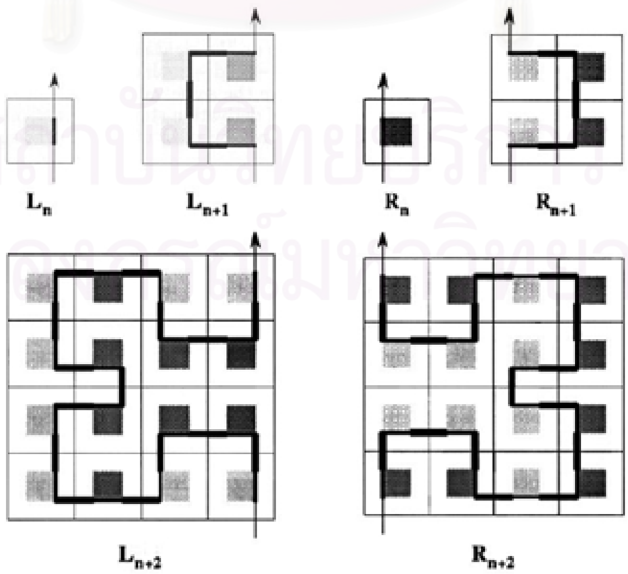


Figure 3.11: Recursive construction of the Hilbert curve in term of replacement.

The curves are generated by either edge or node rewriting method. As in the case of edge rewriting, the relationship between node rewriting and tilings of the plane extends to branching structures. It offers a method for synthesizing L-systems that generate objects with a given recursive structure, and links methods for plant generation based on L-systems.

3.8 Parametric L-systems

Although L-systems with turtle interpretation make it possible to generate a variety of interesting objects, from abstract fractals to plant-like branching structures, their modeling power is quite limited. A major problem can be traced to the reduction of all lines to integer multiples of the unit segment. As a result, even such a simple figure as an isosceles right-angled triangle cannot be traced exactly, since the ratio of its hypotenuse length to the length of a side is expressed by the irrational number $\sqrt{2}$.

Rational approximation of line length provides only a limited solution, because the unit step must be the smallest common denominator of all line lengths in the modeled structure. Consequently, the representation of a simple plant module, such as an internode, may require a large number of symbols. The same argument applies to angles. Problems become even more pronounced while simulating changes to the modeled structure over time, since some growth functions cannot be expressed conveniently using L-systems. Generally, it is difficult to capture continuous phenomena, since the obvious technique of discretizing continuous values may require a large number of quantization levels, yielding L-systems with hundreds of symbols and productions. Consequently, model specification becomes difficult, and the mathematical beauty of L-systems is lost.

Parametric L-systems operate on *parametric words*, which are strings of *modules* consisting of *letters* with associated *parameters*. The letters belong to an *alphabet V*, and the parameters belong to the set of *real numbers* $\Re$. A module with letter $A \varepsilon V$ and parameters $a1,a2,\ldots,an \varepsilon \Re$ is denoted by $A(a1,a2,\ldots,an)$. Every module belongs to the set $M = V \times \Re^{*}$, where $\Re^{*}$ is the set of all finite sequences of parameters. The set of all strings of modules and the set of all nonempty strings are denoted by $M^{*} = (V \times \Re^{*})^{*}$ and $M^{+} = (V \times \Re^{*})^{+}$, respectively.

The real-valued *actual* parameters appearing in the words correspond with *formal* parameters used in the specification of L-systems productions. If Σ is a set of formal parameters, then $C(\Sigma)$ denotes a *logical expression* with parameters from Σ, and $E(\Sigma)$ is an *arithmetic expression* with parameters from the same set. Both types of expressions consist of formal parameters and numeric constants, combined using the arithmetic operators +, -, *, /; the exponentiation operator ^, the relational operators <, >, =; the logical operator !,&,| (not, and, or); and parentheses ( ). Standard rules for constructing syntactically correct expressions and for operator precedence are observed. Relational and logical expressions evaluate to zero for false and one for true. A logical statement specified as the empty string is assumed to have value one. The sets of all correctly constructed logical and arithmetic expressions with parameters from Σ are noted $C(\Sigma)$ and $\varepsilon(\Sigma)$.

A *parametric OL-system* is defined as an ordered quadruplet $G = \langle V, \Sigma, \omega, P \rangle$, where

- V is the *alphabet* of the system,
- Σ is the *set of formal parameters,*
- $\omega \; \varepsilon \; (V \times \Re^{*})^{+}$ 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*.

The symbols: and → are used to separate the three components of a production: the *predecessor*, the *condition* and the *successor*. For example, a production with predecessor $A(t)$, condition $t > 5$ and successor $B(t+1)CD(t\text{^}0.5,t-2)$ is written as

$$A(t) : t > 5 \rightarrow B(t+1)CD(t\text{^}0.5,t-2). \qquad (3.1)$$

A production *matches* a modules in a parametric word if the following conditions are met:

- the letter in the module and the letter in the production predecessor are the same,
- the number of actual parameters in the module is equal to the number of formal parameters in the production predecessor, and
- the condition evaluates to *true* if the actual parameter values are substituted for the formal parameters in the production.

A matching production can be *applied* to the module, creating a string of modules specified by the production successor. The actual parameter values are substituted for the formal parameters according to their position. For example, production (3.1) above matches a module $A(9)$, since the letter A in the module is the same as in the production predecessor, there is one actual parameter in the module $A(9)$ and one formal parameter in the predecessor $A(t)$, and the logical expression $t>5$ is true for $t=9$. The result of the application of this production is a parametric word $B(10)CD(3,7)$.

If a module a produces a parametric word χ as the result of a production application in an L-system G, we write $a \rightarrow \chi$. Given a parametric word $\mu = a1a2\ldots am$, we say that the word $\nu = \chi 1\chi 2\ldots\chi m$ is *directly derived* from (or *generated* by) μ and write $\mu \rightarrow \nu$ if and only if $a_i \rightarrow \chi_i$ for all $i = 1,2,..,m$. A parametric word ν is generated by G in a *derivation of length n* if there exists a sequence of words $\mu_0,\mu_1,\ldots,\mu_n$ such that $\mu_0 = \omega$, $\mu_n = \nu$ and $\mu_0 \rightarrow \mu_1 \rightarrow \ldots \rightarrow \mu_n$.

An example of a parametric L-system is given below.

$$
\begin{array}{llll}
w & : B(2)A(4,4) & & \\
p_1 & : A(x,y) & : y <= 3 & \rightarrow A(x * 2, x + y) \\
p_2 & : A(x,y) & : y > 3 & \rightarrow B(x)A(x/y, 0) \\
p_3 & : B(x) & : x < 1 & \rightarrow C \\
p_4 & : B(x) & : x >=1 & \rightarrow B(x - 1)
\end{array}
\qquad (3.2)
$$

As in the case of non-parametric L-systems, it is assumed that a module replaces itself if no matching production is found in the set P. The words obtained in the first few derivation steps are show in Figure 3.10

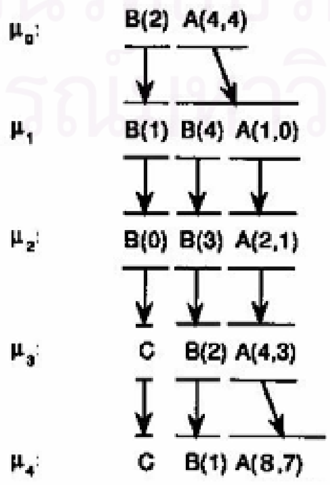


Figure 3.12: The initial sequence of strings generated by the parametric L-system specified in prototype (3.2).

3.9 Turtle interpretation of parametric words

If one or more parameters are associated with a symbol interpreted by the turtle, the value of the first parameter controls the turtle's state. If the symbols are not followed by any parameter, default values specified outside the L-system are used as in the non-parametric case. The basic set of symbols affected by the introduction of parameters is listed below.

Table 3.4 The symbol of parametric words.

Parametric words	Meaning
F(*a*)	Move forward a step of length $a > 0$. The position of the turtle changes to (x', y', z'), where $X' = x + aX_x$ $Y' = y + aX_y$ $Z' = z + aX_z$. A line segment is drawn between points (x, y, z) and (x', y', z').
f(*a*)	Move forward a step of length ***a*** without drawing a line.
+(*a*)	Rotate about Z-axis by an angle of *a* degrees. If ***a*** is positive, the turtle is turned to the left and if ***a*** is negative, the turn is to the right.
&(*a*)	Rotate about Y-axis by angle of ***a*** degrees. If ***a*** is positive, the turtle is pitched down and if ***a*** is negative, the turtle is pitched up.
/(*a*)	Rotate about X-axis by an angle of ***a*** degrees. If ***a*** is positive, the turtle is rolled to the right and if ***a*** is negative, it is rolled to the left.

It should be noted that symbols +, &, and / are used both as letters of the alphabet *V* and as operators in logical and arithmetic expressions. Their meaning depends on the context.

Chapter 4

Plant Module and Experimental Design

This chapter describes the structure of prototype, the L-system prototype, the qualitative model, the L-system string and plant interpretation, the data collection, the growth function, the quantitative model, the visualization, and the model evaluation. The plant module is based on the bracketed L-systems, which represent the structure of plant.

4.1 Structure of Prototype

The biological data from the actual plant observation are very important for simulating the plant development. A flow diagram of a prototype of simulation and visualization of the plant growing is shown in Figure 4.1.

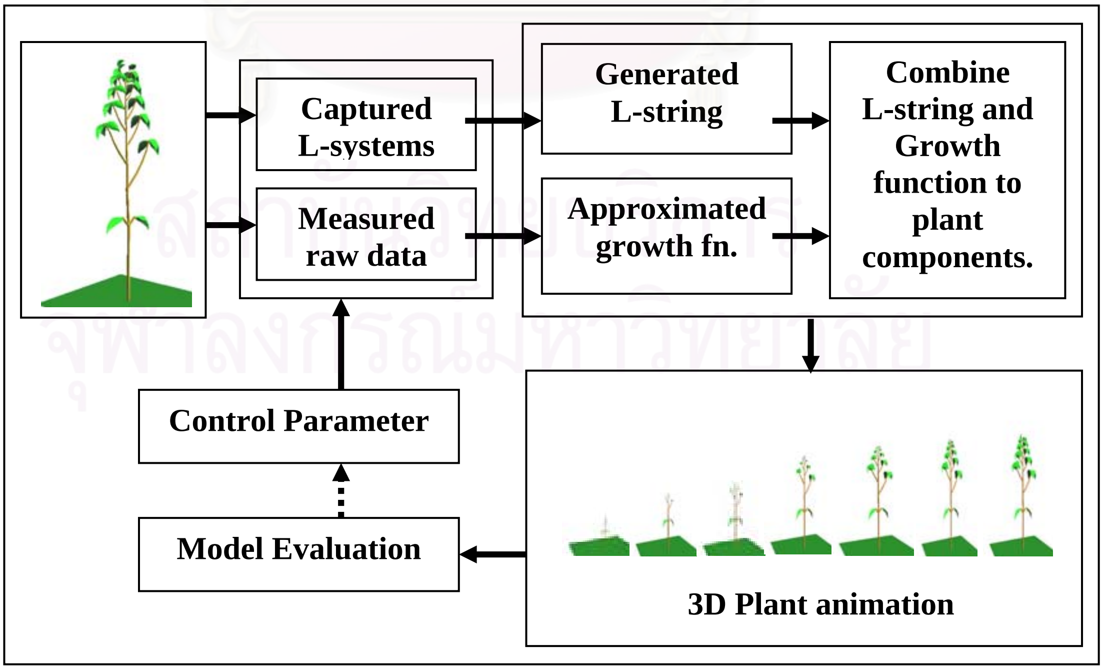


Figure 4.1: Diagram of plant simulation and visualization.

This section presents a prototype for creating computer models that capture the development of plants using L-systems and mathematical model incorporating biological data. The L-system is used for qualitative model in order to represent the plant topology and development. This method has six steps, (1) defining a qualitative model constructed from observations of plant growth in their life cycle, (2) measuring key characteristics collected from actual plants, (3) converting raw data to growth functions based on sigmoidal function approximations, (4) defining a quantitative model composed from the qualitative model and growth function, (5) visualizing the quantitative model, and (6) evaluating the models. The design of L-system in this thesis is based on bracketed L-systems.

4.2 L-system Prototype

The design of L-system in this thesis is based on bracketed L-systems. The symbols are used to represent the plant components. All of these symbols are described in Table 4.1.

Table 4.1: Symbols used in plant growth L-system.

Symbols	Meaning
I	To generate the plant internodes
i	To generate the plant short internodes
P	To generate the plant petioles
p	To generate the plant short petioles
A	To generate the plant apices
L	To generate the plant leaves
F	To generate the plant flowers
+	Roll counter-clockwise to positive Z-axis by angle δ_z, using rotation matrix $R_z(\delta_z)$
-	Roll clockwise to positive Z-axis by angle δ_z, using rotation matrix $R_z(-\delta_z)$
&	Roll counter-clockwise to positive Y-axis by angle δ_y, using rotation matrix $R_y(\delta_y)$
^	Roll clockwise to positive Y-axis by angle δ_y, using rotation matrix $R_y(-\delta_y)$
\	Roll counter-clockwise to positive X-axis by angle δ_x, using rotation matrix $R_x(\delta_x)$
/	Roll clockwise to positive X-axis by angle δ_x, using rotation matrix $R_x(-\delta_x)$
|	Roll back, using rotation matrix $R_y(180)$

Symbols	Meaning
[	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

The simple L-system string is

I [-I] I [+I] I [/I] I [\I] I I   where $\delta_x = \delta_y = \delta_z = 70$, and $\alpha_x = \alpha_y = \alpha_z = 0$.

This string has ten internodes, six for main stem, and four for petiole in four directions. The visualized image is shown in Figure 4.2. The matrices $R_z(-\delta_z)$, $R_z(\delta_z)$, $R_x(-\delta_x)$, and $R_x(\delta_x)$ are used for calculating the symbol string [-I], [+I], [/I], and [\I], respectively.

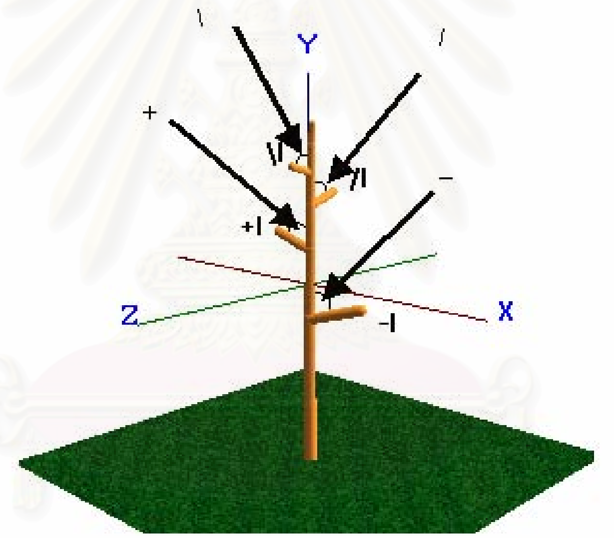


Figure 4.2: A simple L-system Interpretation.
I [-I] I [+I] I [/I] I [\I] I I.

The L-system description of a plant consists of a set of iterations, a set of directional and sizing parameters, initial string, a set of production rules, and a set of terminating productions. The statement

Plant_Name {
    Iterations=N
    Angle=δ
    Diameter=D
    Axiom=ω
    Production 1

Production 2
…
Production *n*
Endrule
Endproduction 1
Endproduction 2
…
Endproduction *m*
}

is the format of L-system.

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=δ

This angle(δ) is used to set the angle of a the branch. For example, '-' is to turn right by an angle δ, '^' is to pitch up by an angle δ, and '/' is to roll right by an angle δ degree.

Diameter=D

This diameter is used to set the diameter of the first internode. The other diameter of internodes and petioles are set follow by the equation of first internode. The unit of *D* is centimeters.

Axiom=ω

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 three-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=Successor

The *predecessor* is a symbol and the *successor* is a symbol string. The symbol is a set of character, and the symbol string is a set of character string.

Endrule

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

Endproduction 1 ... Endproduction m

The format of *endproduction* is the same as the production. The *endproduction* is a symbol that used to specify special plant symbols. These symbols are not defined in Table 4.1.

Character "{" and "}"

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

4.3 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 thesis, the qualitative parameters are obtained 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 4.3 represents an internode *I*, a pair of leave *L* with their short internode *i*, and an apex *A*. The apex *A* is 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.

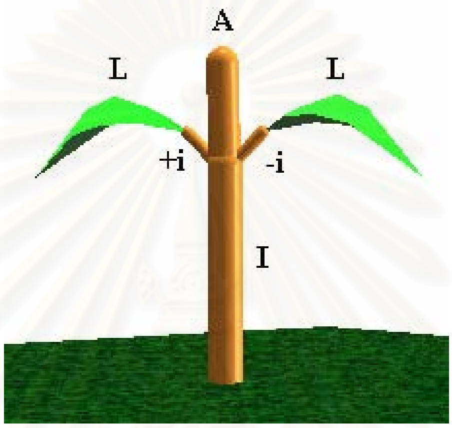


Figure 4.3: Axiom = I[+iL][-iL]A.

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), the right petiole (P) as well as the apex (A) are substituted by an internode and a leaf. By using the previous 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.

The last symbol string at the sixth iteration is

I[+iL][-iL]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]**A**

The string *P* and *B* are not appear in the last string of the above symbol string, but *A* remains in the last letter. *A* is replaced with *endproduction A=IL*. Observe that the *endproduction B=IL* and *P=IL* is not necessary to define in the soybean module. After the above string is replaced with the *endproduction*, the final string is

I[+iL][-iL]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]**IL**

After the sixth iteration and the *Endrule* is substituted to the last symbol string, the symbol string is interpreted by the turtle's interpretations of the L-systems definition. This string is transformed to the associated plant structure with the growth function in each time step.

4.4 L-system String and Plant Interpretation

The last string of generated L-system is interpreted as the plant structure which consists of internodes, petioles, leaves, apices, and flowers that compose to the main stem. The branches is defined from the bracket symbols “[” and “]”. The example of the L-system prototype is given in an example 2.

The simplest L-system is given for understanding the L-system code and L-system interpretation in *Plant*2 prototype.

```
Plant2{
    Iterations=1
    Angle=45
    Diameter=2
    Axiom=IA
    A=[+pL][-pL]I[-I1][+I1]I[/I2][\I2]IF
    1=[/iL][\iL]P
    2=[-iL][+iL]P
    Endrule
    P=iiF
}
```

The *Plant2* prototype has one iteration, the angle of 45 degrees, and two units of diameter. An axiom consists of the internode *I* and the apex *A*. At the first iteration, the apex *A* is substituted by left leaf short petiole and leaf [+pL], right short petiole and leaf [-pL], internode *I*, left petiole and special symbol [-I1], right petiole and special symbol [+I1], internode *I*, back petiole and special symbol [/I2], front petiole and special symbol [\I2], internode *I*, and flower *F*. The second production, a symbol 1 is replaced by back short internode and its leaf [/iL], front short internode and its leaf [\iL], and the petiole *P*. The third production, a symbol 2 is replaced by short right petiole and its leaf [-iL], short left petiole and its leaf [+iL], and the petiole *P*. The *endproduction P* is substituted by two short internode and flower for every petiole *P* in the last L-system string. After the first iteration, the last string is given below.

```
I[+pL][-pL]I[-I[/iL][\iL]iiF][+I[/iL][\iL]iiF]I[/I[-iL][+iL]iiF][\I[-iL][+iL]iiF]IF
```

The internode order, leaf order, and flower order of the L-system string are described in Figure 4.4. The internode order is counted from the first to the last L-system symbol string for symbol *I*, *i*, *P*, and *p*. The leaf (L) and flower (F) order are attached to the previous internode or petiole. Every internode and petiole has an attribute for leaf and flower. The attribute of internode is described in Section 4.8.

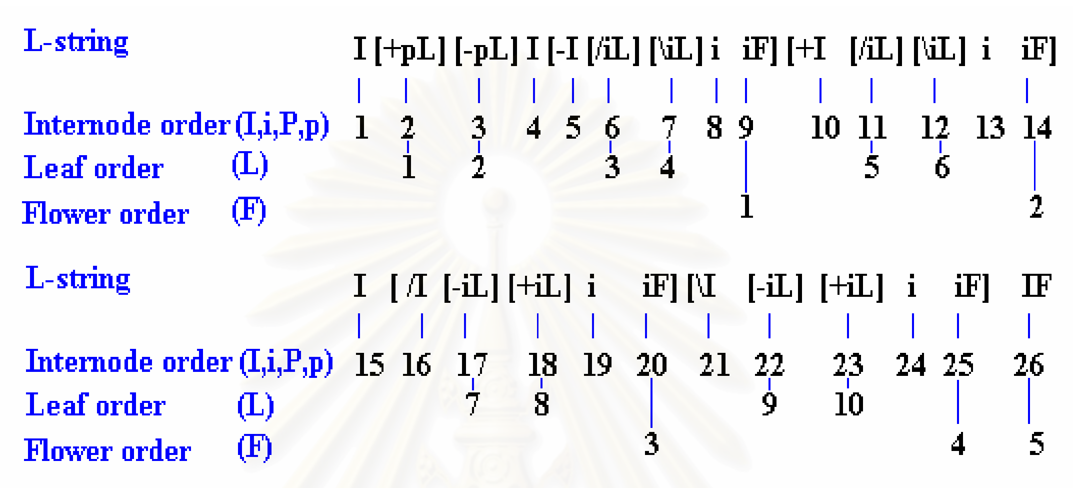


Figure 4.4: The component order of L-system string.

The visualization of the L-system string is presented in Figure 4.5. It shows only the internodes and petioles. The previous nearest component of the internode or the petiole is the parent of component *i*. Figure 4.6 shows the number of internode order.

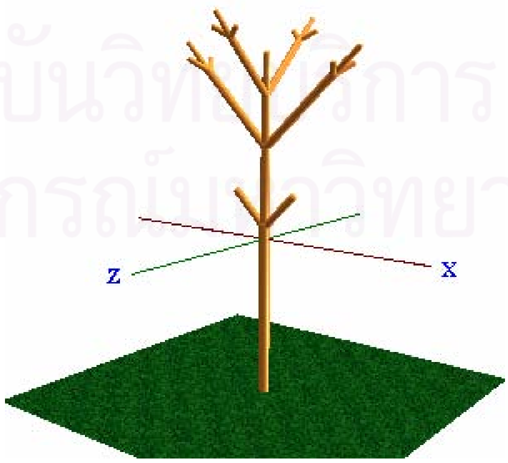


Figure 4.5: All internodes and petioles.

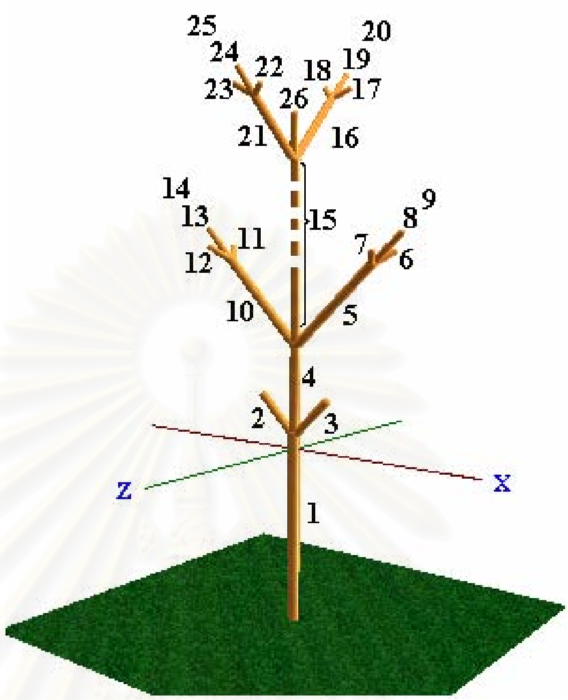
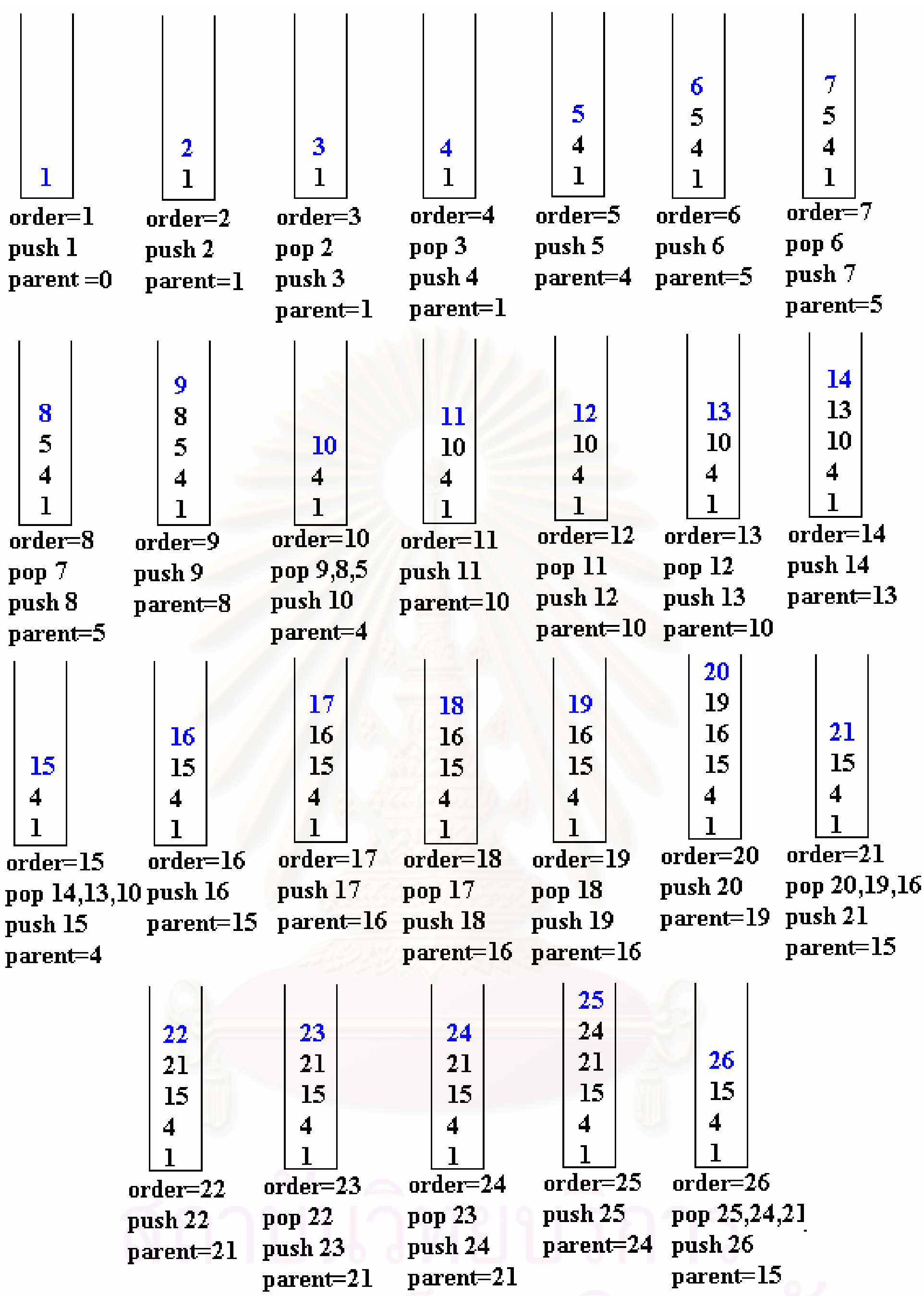


Figure 4.6: All internodes number.

In Figure 4.6, the number of each internode is counted following the L-system string as in Figure 4.4. It is used to determine the parent of each component. The number of symbol [in the last L-system string is used to generate the main stem of plant. The level of each component is used to calculate the initial time of each component. The number of bracket [is calculated from the following equation.

Number of Bracket [= (Number of symbol [ ) – (Number of symbol])

The parent is used to set the attribute of each component in the visualization process, for the example in Figure 4.6, the internode 1 is the parent of the internode 2, 3 and 4. The parent value of component can be calculated from the number of bracket [and the number of main internode in the current state. Each stack of every state with the order number of internode or petiole of the internode *I*, short internode *i*, petiole *P*, or short petiole *p* are shown in Figure 4.7.



I[+pL][-pL]I[-I[/iL][\iL]iiF][+I[/iL][\iL]iiF]I[/I[-iL][+iL]iiF][\I[-iL][+iL]iiF]IF

Figure 4.7: The parent of each internode and petiole.

At the last order of Figure 4.7, the main stem of the plant will be arranged in the stack. For example, the main stem of this example is the first, the fourth, the 15th, and the 20th internode, respectively. The level of each component is considered from the order of the parent calculated from the following equation:

Level of component i (L_i) = (Level of parent component i) + 1

where the level of the parent component 0 is zero. Figure 4.8 shows the level of the plant. It is used to set the initial time of its component. It will be shown in Section 4.5. The level of component, the main stem order, the parent of component, and the number of bracket [are shown in Table 4.2.

where
L_i : Level of component
M_i : Main stem order
P_i : Parent of component
N([) : Number of bracket [

Table 4.2: The value of leaf number, flower number, number of [, parent of each component, main stem order, and level of each component.

Component number *i*	Symbol	Leaf number	Flower Number	N([)	P_i	M_i	L_i
1	I	-	-	0	0	1	1
2	p	1	-	1	1	-	2
3	p	2	-	1	1	-	2
4	I	-	-	0	1	2	2
5	I	-	-	1	4	-	3
6	i	3	-	2	5	-	4
7	i	4	-	2	5	-	4
8	i	-	-	1	5	-	4
9	i	-	1	1	8	-	5
10	I	-	-	1	4	-	3
11	i	5	-	2	10	-	4
12	i	6	-	2	10	-	4
13	i	-	-	1	10	-	4
14	i	-	2	1	13	-	5
15	I	-	-	0	4	3	3
16	I	-	-	1	15	-	4
17	i	7	-	2	16	-	5
18	i	8	-	2	16	-	5
19	i	-	-	1	16	-	5
20	i	-	3	1	19	-	6
21	I	-	-	1	15	-	4
22	i	9	-	2	21	-	5
23	i	10	-	2	21	-	5
24	i	-	-	1	21	-	5
25	i	-	4	1	24	-	6
26	I	-	5	0	15	4	4

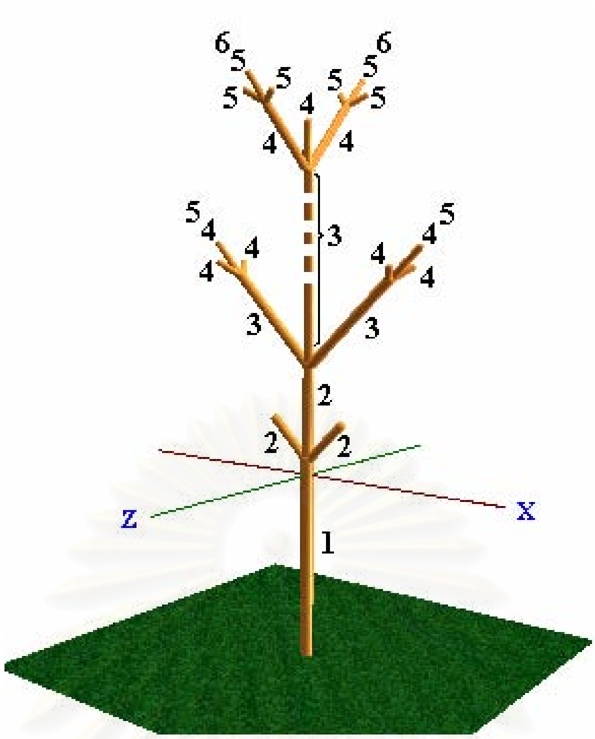


Figure 4.8: The level of each component.

From this prototype, the leaf *L* and the flower *F* are added to the system and the visualization is shown in Figure 4.9.

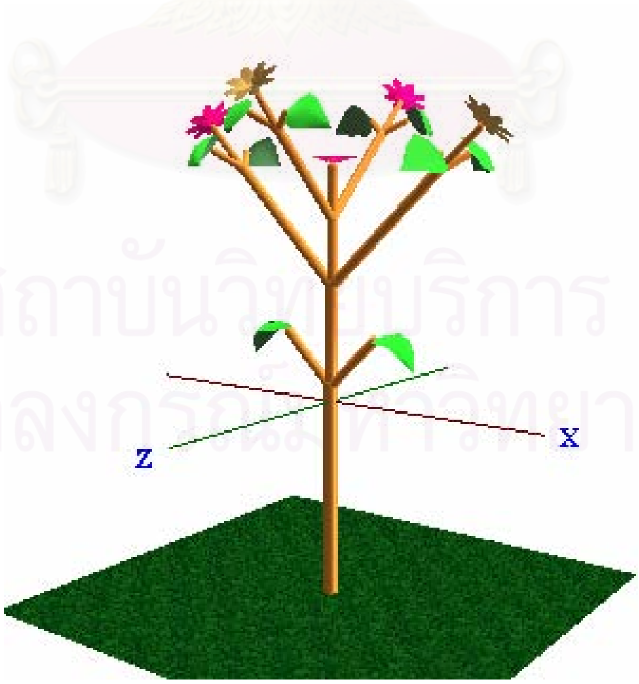


Figure 4.9: The visualized image after adding leaf and flower to the system.

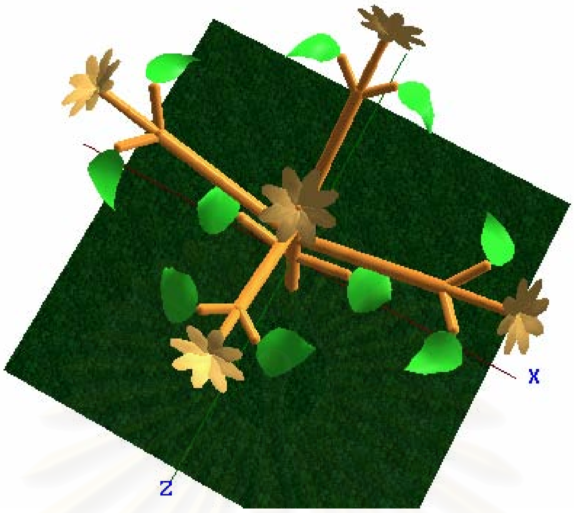


Figure 4.10: The top view of *Plant2* prototype.

In the case of soybean, the L-system string is more complicated, so we reduced the process to two iterations. The last L-system string is given below.

I[-iL][+iL]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]IL

The internodes and petioles of the soybean model are shown in Figure 4.11, and the number of internode and petiole are shown in Figure 4.11. The level of each component and its parent is shown in Figure 4.13 by following the concept of the *Plant2* prototype. After adding the leaf *L* to the plant, the visualization is shown in Figure 4.14, and the top view of its topology is expressed in Figure 4.15.

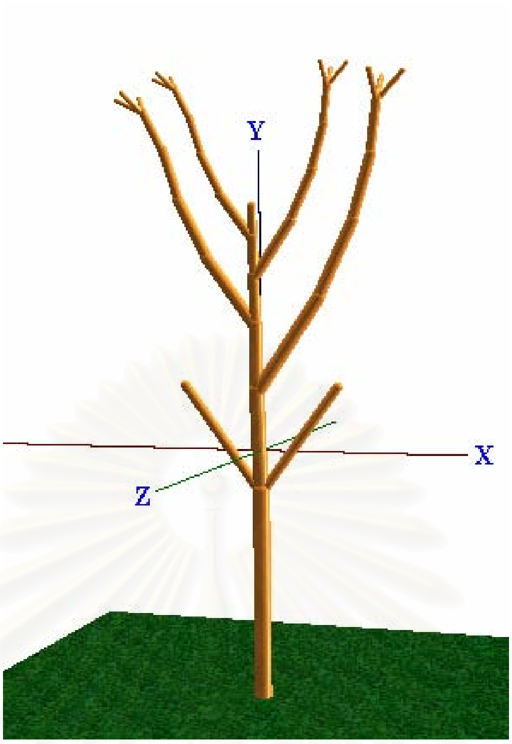


Figure 4.11: The internode and petiole of soybean at second iteration.

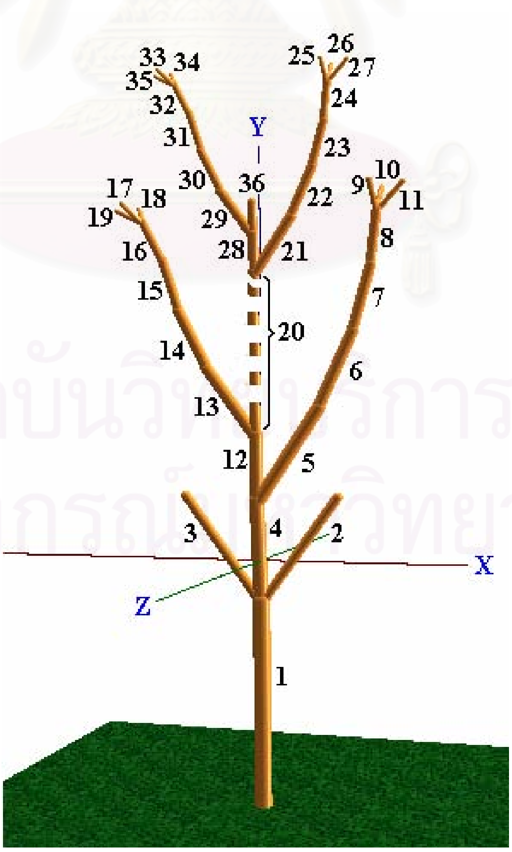


Figure 4.12: The internode and petiole order number.

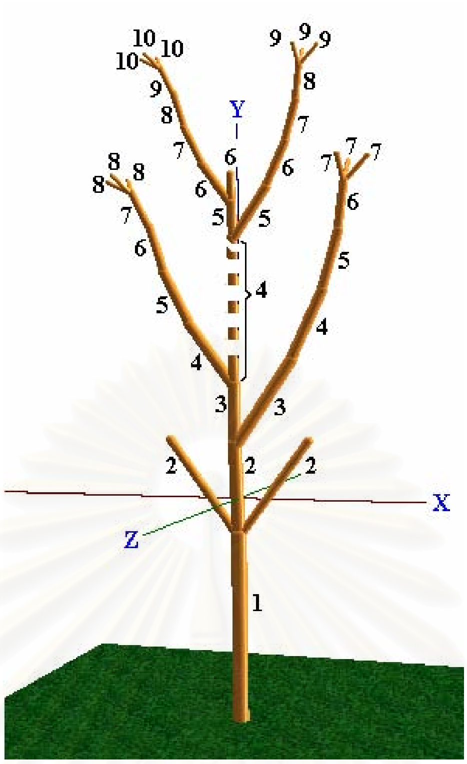


Figure 4.13: The level of all internode and petiole.

Figure 4.14: The soybean and its leaves.

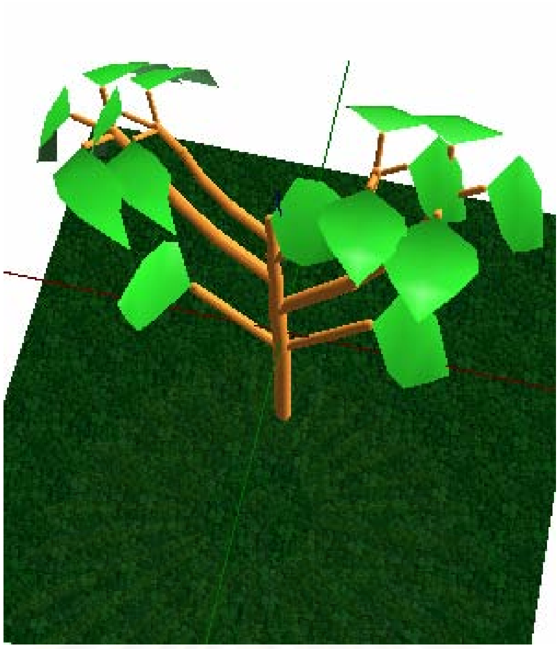


Figure 4.15: The top view of the soybean.

The structure of rewriting algorithm is similar to Pascal. The structure of the production rule consists of the predecessor and successor which is given below.

```
Type production = record
begin
        Pred    : Character
        Succ    : String
end
```

The variables are defined as follows:

i, j, and k	: a positive integer,
iter	: a number of the L-system iterations,
rulenum	: a number of production rules,
allstr	: a last L-system string,
rule	: an array of the production rule,
prev	: a previous string of current character,
last	: a next string of current character,
Endrule_Check	: a flag of the endrule keyword, it is true if there are endrule keyword,
Length()	: a length function return the number of argument string,
EndRuleNum	: a number of endproduction rules,
Endrule	: an array of the production rule,

The rewriting algorithm of the L-system is given below.

Rewriting Algorithm

```
BEGIN
  FOR i := 1 TO iter DO
  BEGIN
        FOR j := 1 TO rulenum DO
        BEGIN
              k := 1
              WHILE (k <= length(allstr)) DO
              BEGIN
                    IF  allstr[k]=rule[j].pred THEN
                    BEGIN
                          prev := Copy(allstr,1,k-1)
                          last := Copy(allstr,k+1,length(allstr)-k)
                          allstr := prev + rule[j].succ + last
                          k := k+length(rule[j].succ)
                    END
                    ELSE  k:=k+1
              END
        END
  END

  IF EndRule_Check THEN
  BEGIN
        FOR j := 1 TO EndRuleNum DO
        BEGIN
              k := 1
              WHILE (k <= length(allstr)) DO
              BEGIN
                    IF  allstr[k]=EndRule[j].pred THEN
                    BEGIN
                          prev := Copy(allstr,1,k-1)
                          last := Copy(allstr,k+1,length(allstr)-k)
                          allstr := prev + EndRule[j].succ + last
                          k := k+length(EndRule[j].succ)
                    END
                    ELSE  k:=k+1
              END
        END
  END
END.
```

After the the rewriting process is finished, the L-system string is interpreted by the L-system interpretation algorithm, the structure and variables are defined as follows:

The structure of plant is given the following type:

```
Type TTree=record
        StringType      : Char;
        Length          : Real;
        BigT            : Real;
        T               : Real;
        t_start,t_stop  : Real;
        Angle_H, Angle_U, Angle_L : Glfloat;
        Angle           : GLfloat;
        angle_azimut    : GLfloat;
        allchild        : Integer;
        child           : array[1..maxchild] of integer;
        myparent        : Integer;
        mylevel         : Integer;
        Leaf            : Boolean;
        Flower          : Boolean
End.
```

The description of TTree attribution is given as follows:

StringType	: a symbol of the L-system string,
Length	: a length of internode, or petiole,
BigT	: a maximum life time of internode or petiole,
t	: a time variable between the internode or petiole life cycle,
t_start,t_stop	: a start time and stop time of the internode or petiole,
Angle_H, Angle_L, Angle_U	: a angle respect to X, Y, Z axis, respectively,
allchild	: a number of all child of internode or petiole,
child	: an array of child internode or petiole,
myparent	: a parent of the internode or the petiole,
mylevel	: a level of the internode or the petiole,
Leaf	: a flag for internode leaf, it is TRUE if there is a leaf,
Flower	: a flag for internode flower, it is TRUE if there is a flower,

The variables are defined as follows:

CountNode	: a number of symbols *I*, *A*, *P*, *i*, and *p*,
Showmessage()	: a procedure that display the argument message,
Exit	: an exiting procedure,
NOT	: a boolean operator,
Maxatree	: a maximum length of the L-system string,
Atree	: an array of plant TTree type,
Maxatree	: a maximum number of symbols *I*, *A*, *P*, *i*, and *p*,
Current_Str[k]	: a k^{th} symbol of the L-system string,
Main_Stem[k]	: an array of the main stem order at k^{th} level,
MainStemOfBracket[k]	: a number of bracket [at the k^{th} level,
Bracket	: a number of Bracket [,
Ang_U	: an angle respect to Z axis,
Ang_L	: an angle respect to Y axis,

Ang_H : an angle respect to X axis
TopStack : a length of stack,
Stack_Angle_U[TopStack] : a Topstack angle respect to Z axis,
Stack_Angle_L[TopStack] : a Topstack angle respect to Y axis,
Stack_Angle_H[TopStack] : a Topstack angle respect to X axis,
RU_Change : a flag of turtle angle on Z axis,
RL_Change : a flag of turtle angle on Y axis,
RH_Change : a flag of turtle angle on X axis,
AllNode : a number of all component *I*, *i*, *A*, *P*, *p*,
Current_Node : a current symbol of the L-system string,
TopMainStem : a current length of the main stem, and
MainStem[TopMainStem] : a main stem order at TopMainStem level.

The L-system interpretation algorithm is given below.

L-system interpretation algorithm

```
BEGIN
        CountNode := 0;
        FOR k := 1 TO length(allstr) DO
        BEGIN
           IF allstr[k] IN ['I','A','P','i','p'] THEN
                 CountNode := CountNode + 1
           IF NOT (allstr[k] in ['I','i','A','P','p','L','F','[',']','-','+','/','\','^','&','|']) THEN
           BEGIN
                 Showmessage(allstr[k]+'is not in {I,i,A,P,p,L,F,[,],/,\,+,-,^,&,|}')
                 EXIT
           END
        END

        IF CountNode > Maxatree THEN
        BEGIN
           Showmessage('The String is too long.')
           EXIT
        END

        FOR k:= 1 TO CountNode DO
        BEGIN
                 atree[k].Angle_U := 0
                 atree[k].Angle_L := 0
                 atree[k].Angle_H := 0
                 atree[k].Leaf := False
                 atree[k].Flower := False
        END

        IF CountNode > MaxaTree THEN
        BEGIN
                 Showmessage('The number of node is too much.')
                 EXIT
        END
```

```
        ELSE
                FOR k:= 1 TO length(allstr) DO
                BEGIN
                        Current_Str[k] := 0
                        Main_Stem[k] := 0
                        MainStemOfBracket[k] := 0
                END

//------------ Plant generator start ------------
        k := 1
        WHILE k<=length(allstr) DO
        BEGIN
                CASE allstr[k] OF
                '[' :   BEGIN
                                Bracket := Bracket + 1
                                MainStemOfBracket[Bracket] := 0

                                Ang_U := 0
                                Ang_L := 0
                                Ang_H := 0

                                Stack_Angle_U[TopStack] := Ang_U
                                Stack_Angle_L[TopStack] := Ang_L
                                Stack_Angle_H[TopStack] := Ang_H

                        END

                ']' :   BEGIN
                            Bracket := Bracket - 1
                            Current_Str[TopStack] := 0  // Clear the top of Stack

                            Ang_U := Stack_Angle_U[TopStack]
                            Ang_L := Stack_Angle_L[TopStack]
                            Ang_H := Stack_Angle_H[TopStack]

                            TopStack := TopStack - MainStemOfBracket[Bracket+1]
                            MainStemOfBracket[Bracket+1] := 0

                            RU_Change := True
                            RL_Change := True
                            RH_Change := True
                        END

        'A','P','I','i','p' :
                        BEGIN
                           AllNode := AllNode + 1
                           TopStack := TopStack + 1
                           Current_Node := CountNode + 1
                           Current_Str[TopStack] := Current_Node
```

```
   MainStemOfBracket[Bracket] := MainStemOfBracket
            [Bracket] + 1

IF (RU_Change) THEN
BEGIN
        IF (allstr[k+1] IN ['I','A','P','i','p']) THEN
        BEGIN
                RU_Change :=True
                atree[Current_Node].Angle_U := Ang_U
        END
        ELSE   RU_Change :=False
END

IF (RL_Change) THEN
BEGIN
        IF (allstr[k+1] IN ['I','A','P','i','p']) THEN
        BEGIN
                RL_Change :=True
                atree[Current_Node].Angle_L  := Ang_L
        END
        ELSE   RL_Change :=False
END

IF (RH_Change) THEN
BEGIN
        IF (allstr[k+1] IN ['I','A','P','i','p']) THEN
        BEGIN
                RH_Change :=True
                atree[Current_Node].Angle_H  := Ang_H
        END
        ELSE   RH_Change :=False
END

IF Bracket = 0 THEN   // Check for MainStem
BEGIN
        inc(TopMainStem)  // = 0 is empty node.
        Main_Stem[TopMainStem] := Current_Node
        IF (allstr[k-1] IN ['[','I','A','P',']','i','p']) THEN
                atree[Current_Node].angle  := 0

        Ang_U  := 0.01*Random(5)
        Ang_L  := 0.01*Random(5)
        Ang_H  := 0.01*Random(5)

        IF allStr[k] = 'I' THEN
                atree[Current_Node].StringType  := 'I'
        ELSE IF allStr[k] = 'i' THEN
                atree[Current_Node].StringType  := 'i'
        ELSE   atree[Current_Node].StringType  := allstr[k]
END
```

```
            ELSE  IF allStr[k] IN ['I','i'] THEN
            BEGIN
            IF allStr[k] = 'I' THEN
                    atree[Current_Node].StringType  := 'P'
            ELSE    atree[Current_Node].StringType  := 'p';

            // Reset the angle of Petiole after first Node

    IF allStr[k-1] IN ['I','i'] THEN
    BEGIN
            IF RU_Change AND (atree[Current_Node1].Angle_U>0)
            THEN
                    Ang_U  := 0.05
            ELSE    Ang_U  := -0.05

            IF RL_Change AND (atree[Current_Node1].Angle_L>0) THEN
                    Ang_L  := 0.05
            ELSE    Ang_L  := -0.05

            IF RH_Change AND (atree[Current_Node1].Angle_H>0) THEN
                    Ang_H  := 0.05
            ELSE    Ang_H  := -0.05
            END
    END
    ELSE
            atree[Current_Node].StringType  := allstr[k]

            // Set Leaf attach to this node
            IF allstr[k+1]='L' THEN
            BEGIN
                    atree[Current_Node].Leaf  := True
                    k :=k+1
            END
            // Set Flower attach to this node
            ELSE IF allstr[k+1]='F' THEN
            BEGIN
                    atree[Current_Node].Flower := True
                    k:=k+1
            END

            atree[Current_Node].myparent := Current_Str[TopStack-1]

            atree[Current_Node].Angle_U := Ang_U
            atree[Current_Node].Angle_L := Ang_L
            atree[Current_Node].Angle_H := Ang_H

            atree[Current_Node].mylevel := Bracket
    END

'-' :   BEGIN
```

```
                Ang_U := Ang_U - sigma
                RU_Change := True
            END

    '+' :   BEGIN
                Ang_U := Ang_U + sigma
                RU_Change := True
            END

    '|' :   EGIN
                Ang_U := Ang_U + 180.0
                RU_Change := True
            END

    '&' :   BEGIN
                Ang_L := Ang_L + sigma
                RL_Change := True
            END

    '^' :   BEGIN
                Ang_L := Ang_L - sigma
                RL_Change := True
            END

    '\' :   BEGIN
                Ang_H := Ang_H + sigma
                RH_Change := True
            END

    '/' :   BEGIN
                Ang_H := Ang_H - sigma
                RH_Change := True
            END
    END
    k := k + 1
  END
END.
```

4.5 Data Collection

The data of each component are collected from an actual soybean. They are the internodes length, diameter, leaves length and width, petioles length corresponding to the time of its life cycle. The actual data are obtained daily from three soybeans for 61 days. The raw data will be used for approximating to the sigmoidal growth function. The data of soybean were collected manually using rulers and a protractor.

The collected data is shown in Appendix B. We ignore all data that are not vegetative state such as underground part, and the reproductive states. The physiology of soybean is shown in Figure 4.16. The internode length, petiole length, leaf length and width are designed as in Figure 4.17, Figure 4.18, and Figure 4.19, respectively.

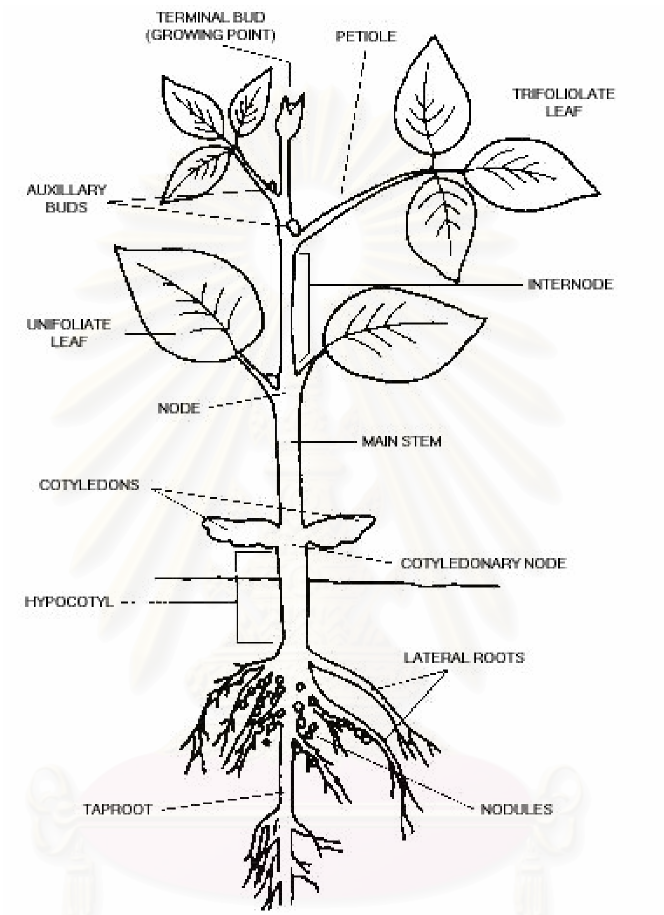


Figure 4.16: The soybean physiology.

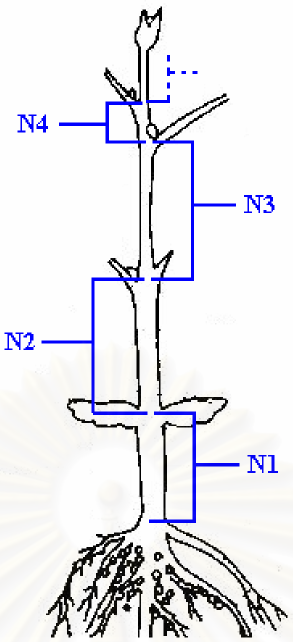


Figure 4.17: Internode data, Ni is the order of internode.

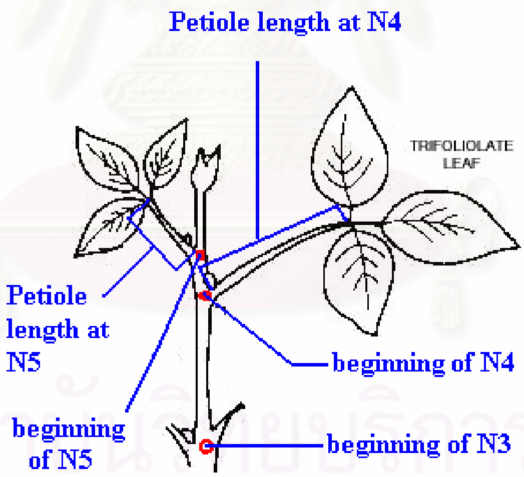


Figure 4.18: Petiole length data.

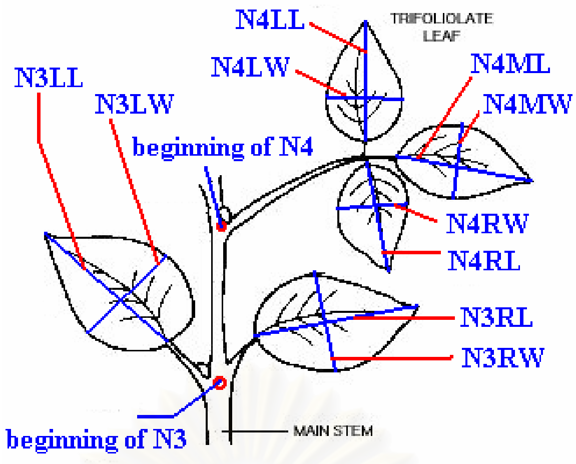


Figure 4.19: Leaf length and width data.

The description of leaf measurement is shown in Figure 4.19. At the beginning of third internode, there are two leaves. The fourth internode and upper internode, there are trifoliolate – three leaves. The symbol description of each leaf is given below.

NiLL : the left leaf length of ith intenode,
NiLW : the left leaf width of ith intenode,
NiRL : the right leaf lenght of ith intenode,
NiRW : the right leaf width of ith intenode,
NiML : the middle leaf length of ith intenode,
NiMW : the middle leaf width of ith intenode,

4.6 Growth Function

The raw data in Section 4.5 is approximated as a sigmoidal growth function in shown Figure 4.20.

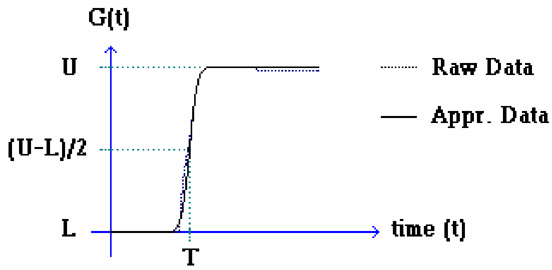


Figure 4.20: Sigmoidal curve approximation.

The raw data is converted to the 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

In the prototype, the system read data from the user interface and compute the approximation function as a sigmoidal curve the algorithm is as the following.

The Simoidal curve approximation Algorithm

1. Read the raw data of length, width, diameter of the internode, the petiole, the leaf, the apex, and the flower corresponding to a time *t*.
2. Compute the minimum value *L* and the maximum *U* from step 1.
3. Compute the time *T* the size (U-L)/2.
4. Compute the appropriate slope of the sigmoidal curve.

   4.1 Compute all error of each slope. The slope *m* will be increased from zero to one with step size 0.01.

```
for i := 0 to 100 do
  begin
     e_i := 0.00;
     m := 0.01*i;
     for j := 1 to n do
     begin
        G(t_j) := L + (U-L)/(1+exp(m*(T-t_j)));
        e_i := e_i + | G(t_j) - y_j |;
     end;
end;
```

   where *n* is a number of day, e_i is an error at value *i*, $G(t_j)$ is and approximated of growth value at time *j*, y_j is a raw data at time *j*, and *L*,*U*,*m* are as the previous

   4.2 Compute the appropriated slope at the minimum error.

```
slope_tmp := e_0;
slope_ok := 0;
for i:= 1 to 100 do
begin
   if slope_tmp > e_i then
   begin
      slope_tmp := e_i;
      slope_ok := i;
   end;
end;
```

Figure 4.21: An algorithm for calculating the simoidal curve.

Besides the growth function, there are other functions, which are used 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

$$Y_i = c(a)^{ni} \qquad \ldots(3)$$

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*.

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

$$B_i = \beta n_i + b \qquad \ldots(4)$$

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.

Every component of plant is controlled by a self-growth function likes the one shown in Figure 4.16 with either same or different slope *m*, time *T*, the maximum *U*, and the minimum *L*.

4.7 Quantitative Model

The quantitative model combines the L-systems string with the approximated growth functions of each component. The plant model can simulate its growth with continuous development in the virtual reality form. In Section 4.3, the qualitative model or the L-system string is defined and generated to the plant structure. In Section 4.6, the approximated growth function is constructed from the raw data corresponding to the time in their life cycle. The slope *m*, the minimum value *L*, the maximum value *U*, and the time *T* are set to its component.

Every component of the plant such as the internode *I*, the petiole *P*, the leaf *L*, the flower *F*, and the apex *A* are controlled by a self-growth function likes the one shown in Figure 4.16 with either same or different slope *m*, the time *T*, the maximum *U*, and the minimum *L*.

For example, the L-system string of *Soybean* prototype in Section 4.3 is shown below:

```
I[-iL][+iL]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]I[-IIII[\pL][/pL][-pL]]I[+IIII[\pL][/pL][+pL]]IL
```

The value of *m*, *L*, *U*, *T* of every component is shown in the Table 4.3

Table 4.3: The value of *L*, *U*, *m*, *T*..

Symbols	*L* value	*U* value	*m* value	*T* value
I and i	0.1275	4.039	0.65	7
P and p	0	6.8737	0.38	8
L width	0	2.7909	0.49	11
L length	0	5.2613	0.54	11
A	0.1275	4.039	0.65	7

The approximated growth function of each component is shown in Figure 4.22. The internode *I* and the short internode *i* are shown in Figure 4.22(a), the petiole *P* and the short petiole *p* are shown in Figure 4.22(b), Figure 4.22(c) shows the leaf length *L* , the leaf width *L* is shown in Figure 4.22(d), and Figure 4.22(e) shows the apex *A*.

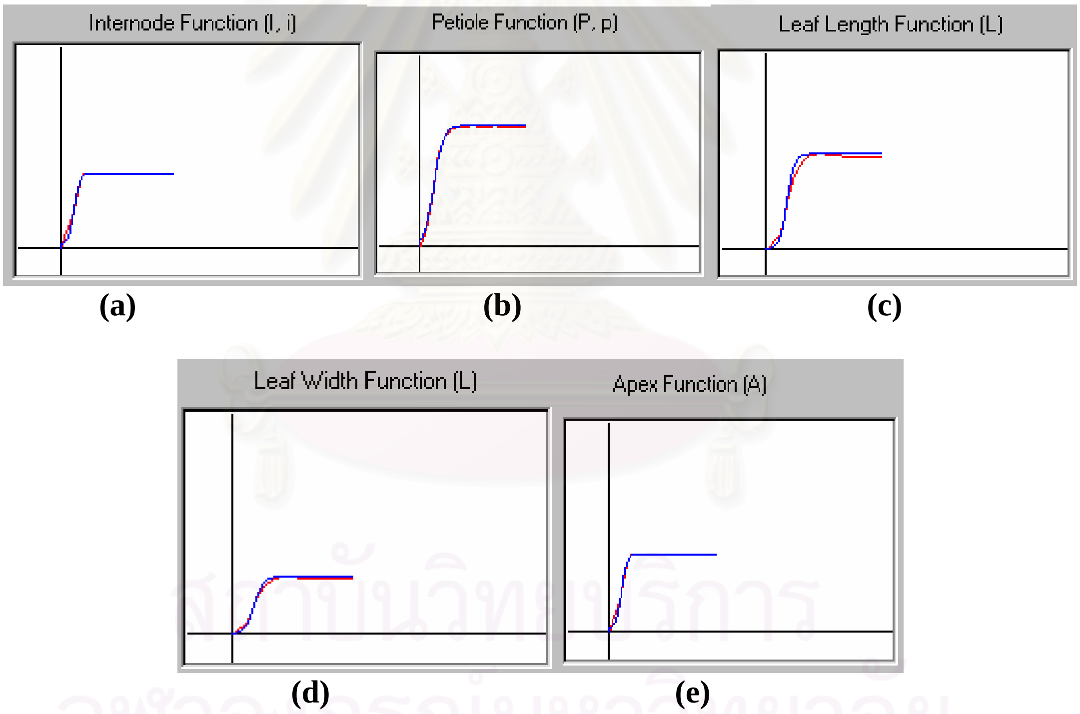


Figure 4.22: The approximated growth function of *I*, *i*, *P*, *p*, *L*, and *A*.

Figure 4.23 illustrates a graphical image of a soybean drawn from the L-system string defined in the production rules in Section 4.3. Figure 4.23(a) shows the L-system string obtained after the rewriting process. The plant structure in Figure 4.23(b) is constructed from the L-system string shown in Figure 4.23(a) and its graphical image of the axiom of the plant is illustrated in Figure 4.23(g).

Figure 4.23(d) gives the details of a part of soybean. Figure 4.23(c) is the left petiole component. The right petiole is shown in Figure 4.23(e). In Figure 4.23(f), the raw data is converted to the growth function corresponding to each symbol.

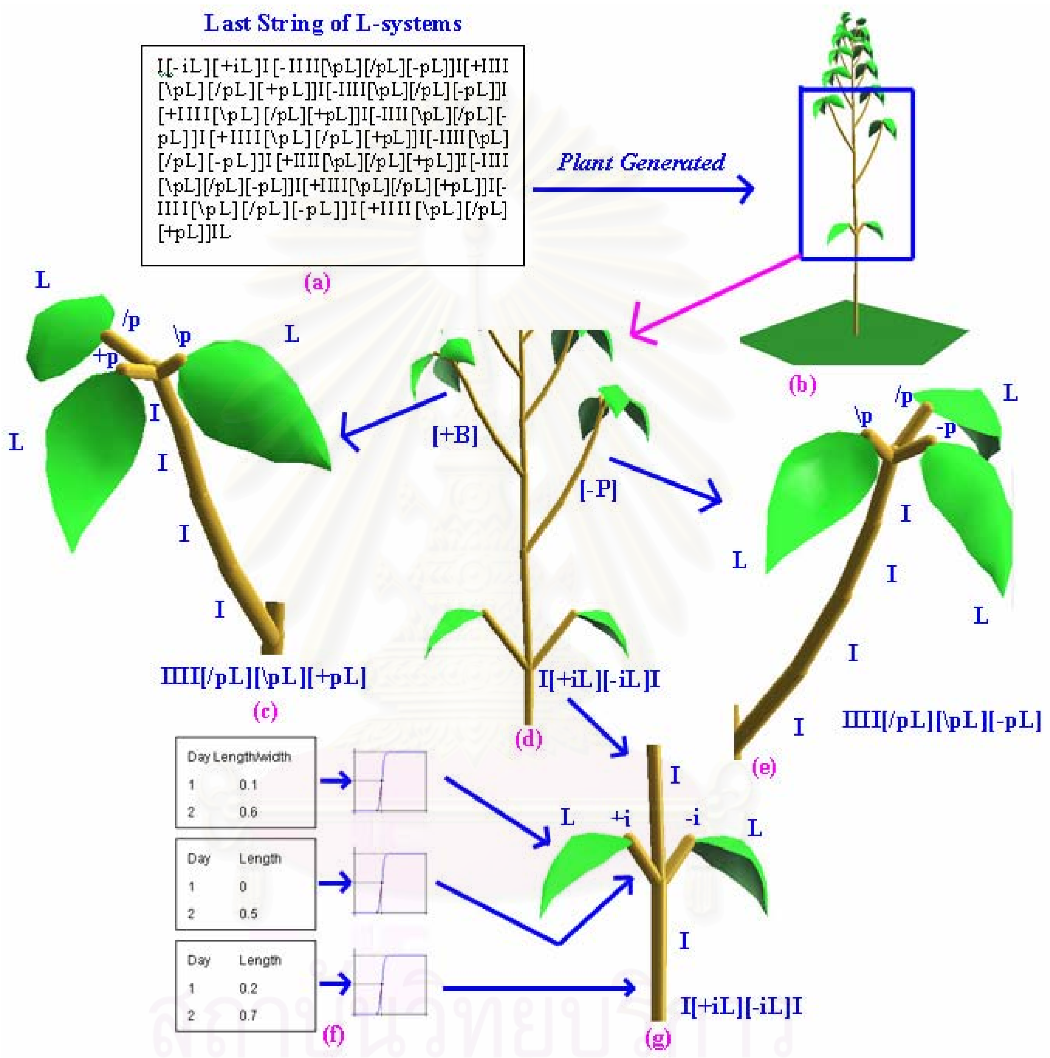


Figure 4.23: Structure of simulation generated from the production rules of soybean in Section 4.3.

4.8 Visualization

This section describes the visualization method of each component such as the internode *I*, the short internode *i*, the petiole *P*, the short petiole *p*, the leaf *L*, the apex *A*, and the flower *F*. Each component uses the primitive geometric shape; for example, cylinder, line, triangular polygon, sphere, rectangular polygon, and texture bitmap.

4.8.1 Internode

A cylinder represents the plant internode, while a sphere represents the internode joint. Figure 4.24 shows the internode of plant. The short internode is similar to internode but it is not the same length. The internode and their ground is shown in Figure 4.25.

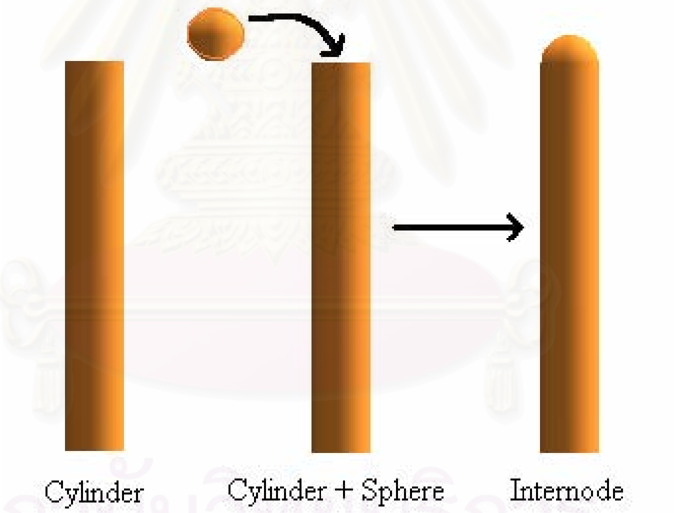


Figure 4.24: Internode of plant topology.

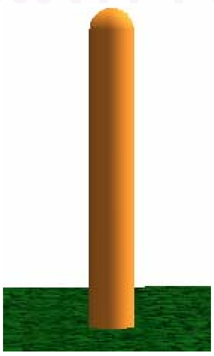


Figure 4.25: Internode plant.

The structure of internode in the prototype is defined by the following type:

```
Procedure InterNode(wx,wy:real)
begin
        Cylinder(wx,wx,wy,slices,stacks)
        Sphere(wx,longitudes,latitudes)
end
```

From above procedure, internode receive 2 arguments, *wx* and *wy* where *wx* represents the base radius of the internode, and *wy* represents the height of the internode. A cylinder has five arguments, namely, base radius, top radius, height, edges, and slices. Spheres have three arguments, namely, radius, latitudes, and longitudes. The cylinder and the sphere are given in Figure 4.26.

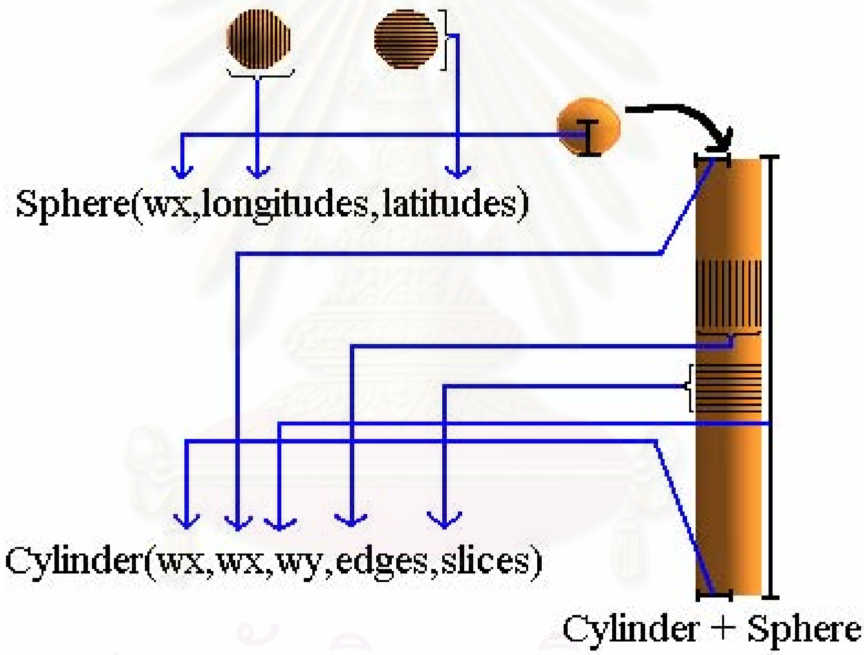


Figure 4.26: The cylinder and sphere argument.

4.8.2 Petiole

The petiole and the main stem have the same topology as internode with either same or different size. The petiole component is represented by the symbol *I*, *i*, *P*, or *p*. Figure 4.27 shows the petiole and the main stem of the plant topology. The short petiole structure is similar to the petiole with either same or different size. The main stem was given in Section 4.4. Its direction is upward to Y-axis.

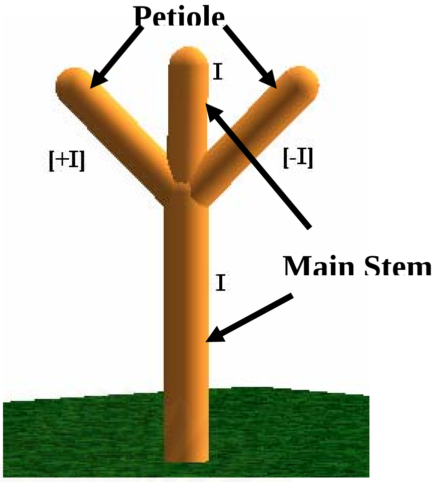


Figure 4.27: The petiole and the main stem of plant topology I[-I][+I]I.

4.8.3 Leaf

A plant leaf is very important for plant visualization to look more realistic. The plant is defined by more than three points in the three-dimensional space. It consists of the *source point* which is attached to the internode or the petiole tip. In this thesis, the leaf library is designed using the following format.

Name=Leaf_Name
Source=S
Point=P_{1x} P_{1y} P_{1z}
Point=P_{2x} P_{2y} P_{2z}
Point=P_{3x} P_{3y} P_{3z}
…
Point=P_{nx} P_{ny} P_{nz}
Triangle=T_{11} T_{12} T_{13}
Triangle=T_{21} T_{22} T_{23}
Triangle=T_{31} T_{32} T_{33}
…
Triangle=T_{m1} T_{m2} T_{m3}

The meaning of each keyword is given as follows:

Name=Leaf_Name

Leaf_Name is the name of leaf.

Source=S

The *source point* is the point in set of the leaf point. It is attached to the tip petiole or the internode that has leaf.

Point=P_{nx} P_{ny} P_{nz}

The keyword is a coordinate (x,y,z) of leaf point. The value P_{nx} , P_{ny} , and P_{nz} are the x, y, z at the point n^{th}. They are a real value.

Triangle=T_{m1} T_{m2} T_{m3}

The triangle consists of three points. The value T_{m1} ,T_{m2} ,T_{m3} are the first point, the second point, and the third point of the m^{th} triangle. They are member in the set of the leaf point.

For example, the leaf prototype is given below.

```
Name=Leaf,Soybean
Source=2
Point=0 0 0
Point=0 0 0
Point=0 0 0
Point=-33 -50 8
Point=0 -50 11
Point=33 -50 8
Point=-44 -100 11
Point=0 -100 17
Point=47 -100 11
Point=-33 -150 5
Point=0 -150 12
Point=33 -150 8
Point=0 -200 0
Point=0 -200 0
Point=0 -200 0
Triangle=1 4 2
Triangle=4 5 2
Triangle=2 5 6
Triangle=2 6 3
Triangle=4 7 5
Triangle=7 8 5
Triangle=5 8 9
Triangle=5 9 6
```

```
Triangle=7 10 8
Triangle=10 11 8
Triangle=8 11 12
Triangle=8 12 9
Triangle=10 13 11
Triangle=13 14 11
Triangle=11 14 15
Triangle=11 15 12
```

From the above illustrates a soybean leaf prototype. The second point is the *source point*. There are 15 points (size 5x3), and 16 triangular polygons. Figure 4.28 shows the soybean leaf topology. At the first, there are 15 points likes Figure 4.28(a) in plane XY coordinate. The first point and the third point are set similar to the second point as same as the 13^{th} point and the 15^{th} point are set to the 14^{th} point. The seventh point is moved leftward while the ninth is moved rightward. The fifth point, the eighth point, and the 11^{th} point are moved upward. The result is shown in Figure 4.28(b). All triangular polygon is set to every triangle in Figure 4.28(c). In Figure 4.28(d), the 13^{th} point, 14^{th} point, and 15^{th} point are moved to downward. It makes the leaf look like longer than the old leaf.

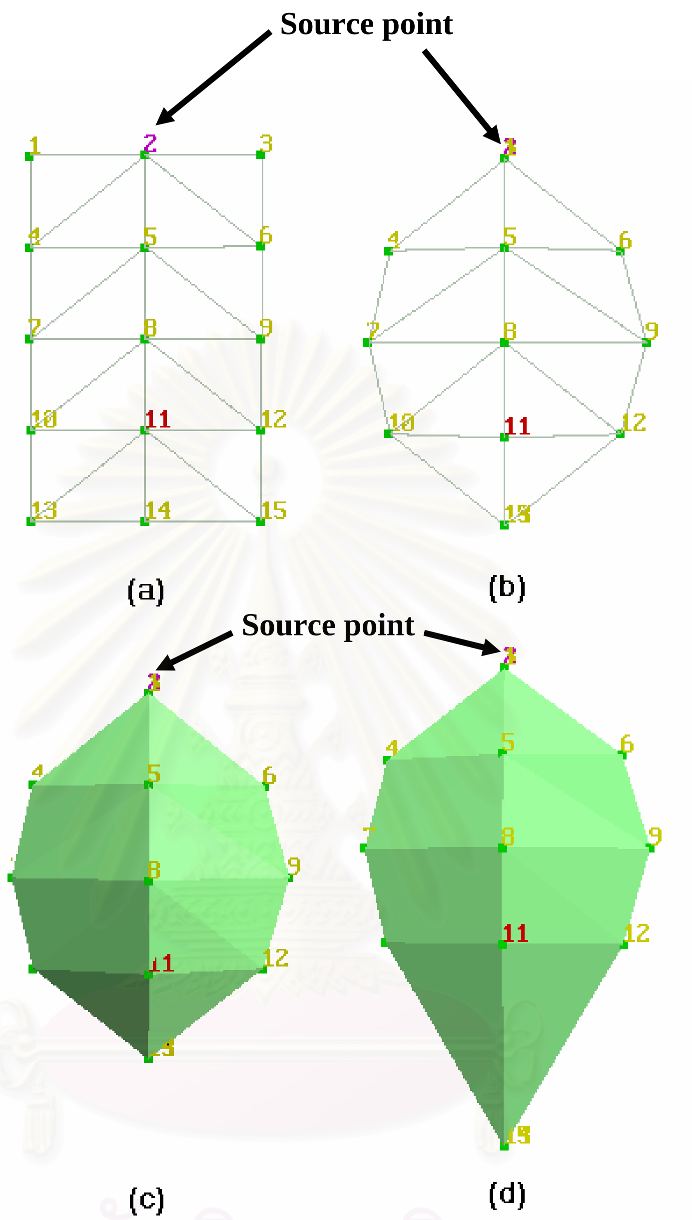


Figure 4.28: The leaf of plant topology.

The leaf grid in this thesis is designed for *J*x*K* points, where *J*, *K* is 3, 5, 7, 9, or 11. The leaf size supports odd number, because this thesis assumes that the leaf is symmetric with respect to the column. A size of leaf depends on its complicated structure.

In Figure 4.29 shows the simple plant with its leaves. The source point of leaf is attached to the internode tip. Therefore the second point is attached to the internode. The leaves in this thesis can be resized and rotated following the user adjustment.

The simple L-system of Figure 4.29 is I[-iL][+iL]A. The previous symbol of the leaf symbol *L* must be only the internode *I*, the short internode *I*, the petiole *P*, or the short petiole *p*. Therefore, the leaf cannot separate from the internode *I*, the short internode *i*, the petiole *P*, or the short petiole *p*.

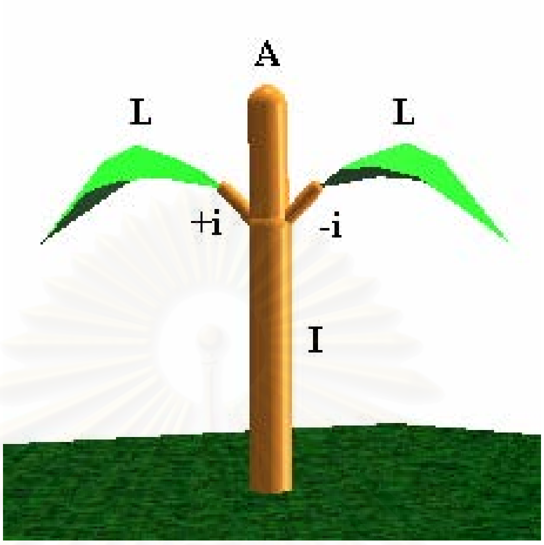


Figure 4.29: A simple plant with its leaves I[-iL][+iL]A.

4.8.4 Flower

A flower of plant in this thesis is defined only rounded flower. The number of petals will be changed by user interface. The petal structure is similar to the leaf structure. It has a source point, set of point, set of triangular polygon. For example, Figure 4.30 shows a simple plant with its leaves and flower with the L-system as follows:

I [-i L] [+i L] I i F

The plant consists of eight components, there are two internodes and one short internode for main stem. There are two petioles, left petiole, and right petiole with their leaf. The flower *F* must follow the internode *I*, the short internode *i*, the petiole *P*, or the short petiole *p*. The symbol *F* is imported from the flower library as well as the leaf library.

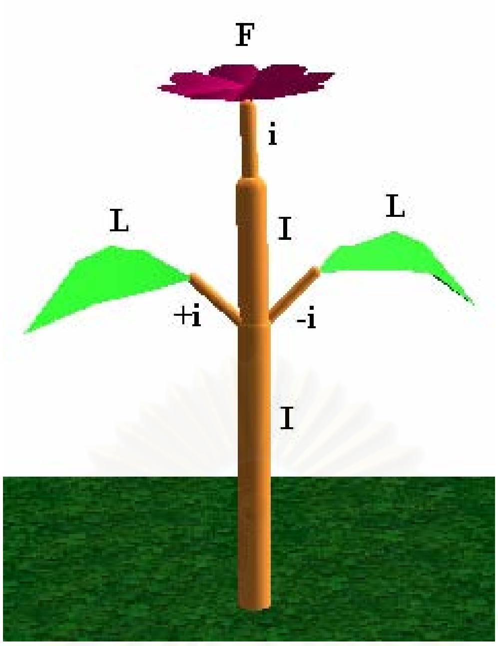


Figure 4.30: A simple plant with its leaves and flower : I[-iL][+iL]IiF.

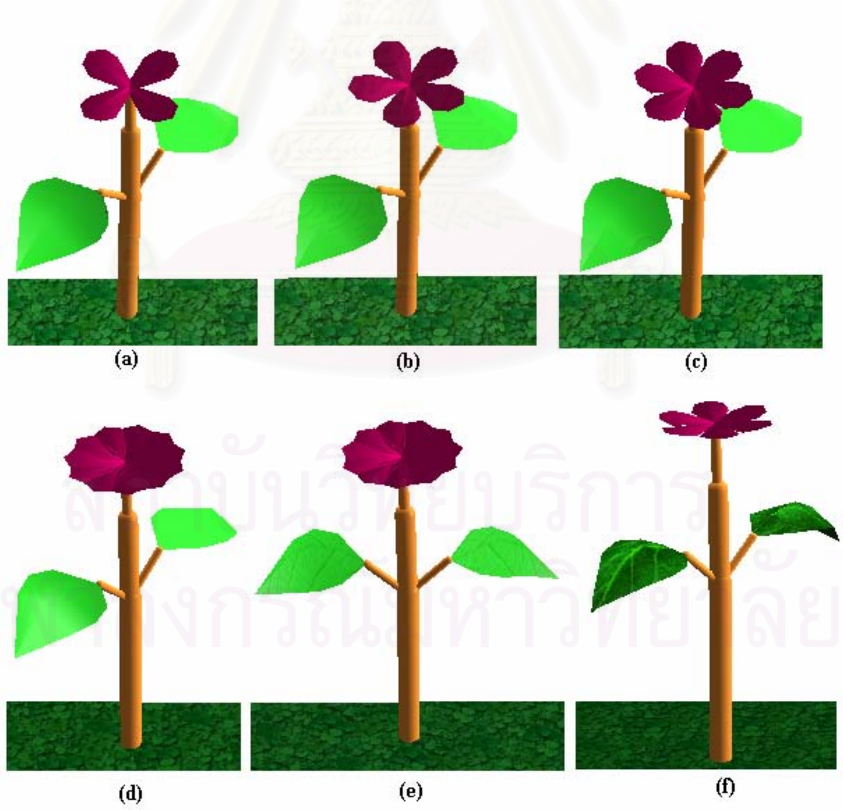


Figure 4.31: Some flower plant topology.

The flower is composed by some petals which user can set the number of petals. It is defaulted at eight petals. In Figure 4.31 shows some flower plant with their leaf. Figure 4.31(a), there are four petals, five petals in Figure 4.31(b), six petals in Figure 4.31(c), 11 petals in Figure 4.31(d), 11 petals in Figure 4.31(e) with leaf texture, and six petals with leaf texture mapping in Figure 4.31(e).

4.8.5 Texture mapping

In this thesis, a texture is only set to leaf component. The texture is designed for 2-dimensional texture mapping to 3-dimensional space of leaf polygon. The texture size must be 2^nx2^m pixels, where *n*, *m* is 3, 4, 5, 6, 7, or 8. It is cropped from the actual leaf and set to the appropriate size. The texture will be changed the color from user adjustment. The user can create the new texture or select from the given texture library.

The texture coordinate is mapped to range of [0,1]. For example, if the texture size is 64x64 pixels, the texture coordinate will map to 1x1. Figure 4.32 shows the mapping coordinate. The texture coordinate (0,0) is mapped to p0 (0,0), (63,0) to p1 (1,0), (0,63) to p2(0,1), (63,63) to p3(1,1), (31,31) to p4(0.5,0.5) as well as other points.

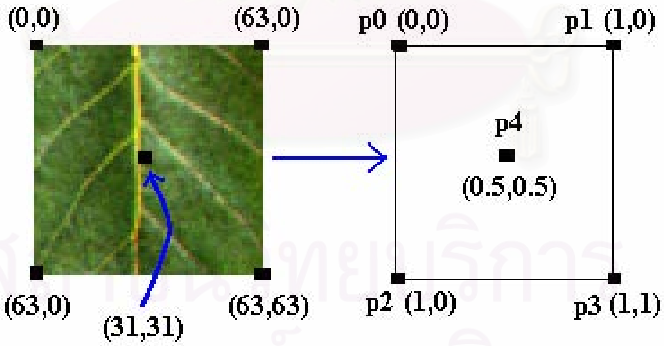


Figure 4.32: The texture coordinate.

Figure 4.33 shows a texture mapping to the leaf polygon. A leaf coordinate in three-dimensional space (Figure 4.33(a)) is mapped to range of [0,1] coordinate in XY plane (Figure 4.33(b)) corresponding to texture coordinate in Figure 4.33(d), and the texture coordinate is mapped to the actual leaf coordinate as Figure 4.33(c).

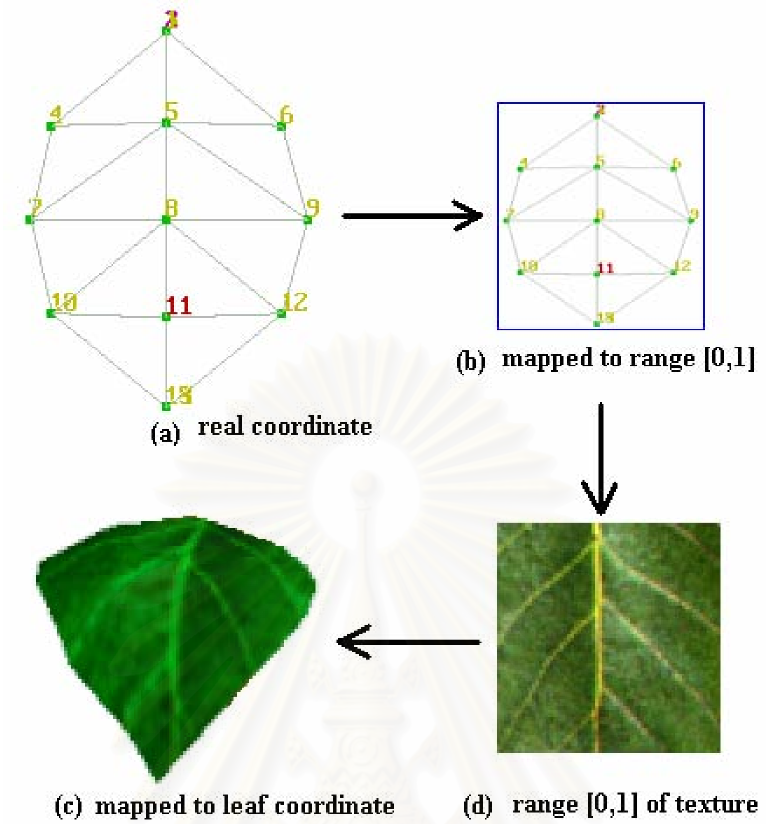


Figure 4.33: A texture mapping with leaf polygon.

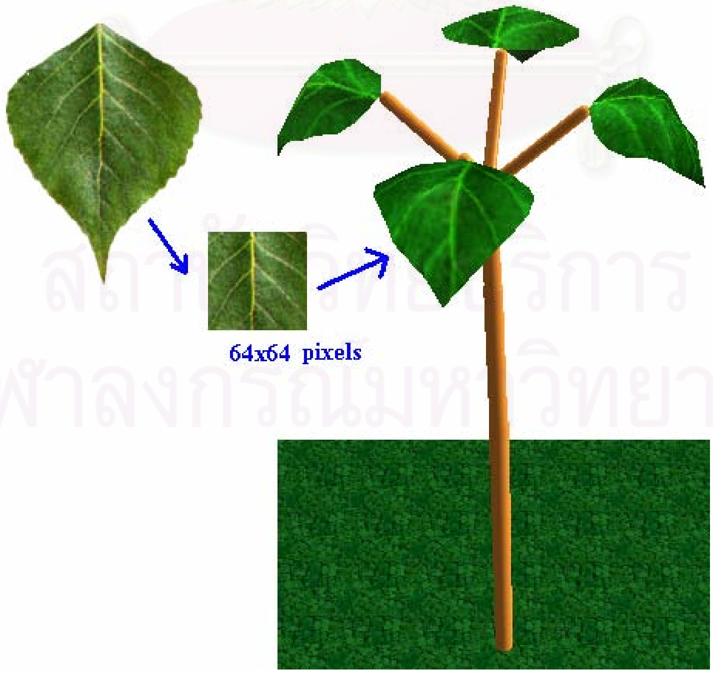


Figure 4.34: The leaf texture mapping.

The flow of leaf texture mapping is shown in Figure 4.34. The texture is scanned by scanner from the actual leaf, and cropped to the appropriate size, then map to the three-dimensional space of leaf point. The three-dimensional leaf can be changed the material color. Figure 4.35 shows a simple plant with its textured leaves and flowers. The L-system string is

I[+iL][-iL]I[-IF][+IF][/IF][\IF]iF

There are two internodes *I*, and a short internode *i* for main stem, a left short internode and its leaf [+iL], a right short internode and its leaf [-iL]. There are four internodes and their flowers in four directions, and a center short internode with its flower. The top view of the simple plant is shown in Figure 4.36.

Figure 4.35: A simple plant with its leaves and flowers.

Figure 4.36: Plant and their component with textured leaves.

In the case of soybean, the texture of the soybean leaf is cropped from the actual soybean in this thesis experiment. It is mapped to the leaf polygon of soybean plant in Figure 4.37.

Figure 4.37: A soybean and textured leaves with *Soybean* prototype using two iterations.

Figure 4.38 visualize the *Soybean* prototype in Section 4.3 using six iterations as well as in Figure 4.39 is added the soybean texture to its leaves.

Figure 4.38: The soybean as follows Section 4.3 using 6 iterations.

Figure 4.39: The soybean as follows Section 4.3 using 6 iterations with textured leaves.

4.9 Model Evaluation

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

Chapter 5

Visualization Procedures and Results

This chapter describes the visualization procedure and the result of the soybean experiment and another plants which is constructed by this prototype.

5.1 Visualization Procedure

The visualization of this thesis consists of three algorithms. There are *Draw scene*, *Draw tree*, and *Perform Node* Algorithm. The *Draw scene* algorithm is the main part. It is *While loop* for plant animating following *GlobalBigT* time variable.

Draw Scene Algorithm

The *DrawScene* algorithm variables is described as following:

GlobalBigT : It is the global time variable of animation.

AnimShow : It is the flag of animation condition. It is true value for showing animation.

DrawLand : The procedure draws the floor of the plant model.

ShowXYZ_Axis : It is the flag of XYZ axis. It is true value to show XYZ axis.

DrawXZ_Axis : The procedure draws the XYZ axis.

DrawTree : The procedure draws the plant corresponding to *GlobalBigT* variable.

The *DrawScene* algorithm is given below.

```
DrawScene
Begin
        1.  GlobalBigT := 0;
        2.  While AnimShow Do
            Begin
                2.1  GlobalBigT := GlobalBigT + 1
                2.2  DrawLand
```

```
            2.3  If showXYZ_Axis then  DrawXYZ_Axis
            2.4  DrawTree(GlobalBigT)
            2.5  If GlobalBigT > Animation Time then
                 GlobalBigT := 1
        End
End.
```

Draw Tree Algorithm

The *DrawTree* algorithm is consisted of two subprocedures, there are *SetTree* for setting the properties of plant component with null value of its child, and *PerformNode* for setting the properties of each component. The argument *bigTime* is the global time time step. The argument 1 of performnode procedure is the initial internode with the first internode of plant.

```
DrawTree(bigTime)
Begin
     1.  SetTree properties and set the initial the child of plant
     2.  PerformNode(1,bigTime)
End.
```

Perform Node Algorithm

The *PerformNode* algorithm is a recurrence procedure. It consists of three steps. First, it computes the properties of each component such as internode, petiole, leaf, apex, flower at the current time *t*. Second, it draws the plant components with the appropriate angle following the L-system symbol string such as internode, petiole, apex, leaf, and flower. Third, the *PerformNode* is called by itself using the current component number and the time *t* to its arguments. The arguments are described below

Treenode	: The current component is initialized by root component. That is the first node.
T	: It is the current time *t*, it is varied from 1 to the global time of the system with the appropriated time step.
Child of Treenode	: It is the child number of *Treenode* component.

```
PerformNode(Treenode, t)
Begin
        1. Compute the properties of each component at current time t
        2. Draw plant components with appropriate angle
            - Internode
            - Petiole
            - Apex
            - Leaf
            - Flower
        3. PerformNode(Child of Treenode, t)
End.
```

5.2 Visualization results

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 5.1 shows some selected stages of the development of a soybean shoot controlled by the production rules defined in Section 4. The developments in Figure 5.1 begin at time *t* = 1 according to the sigmoidal curve in Figure 4.20.

Figure 5.2 shows various plant structures with the same topology of L-system under different parameters. The L-system code of Figure 5.2 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
  }
```

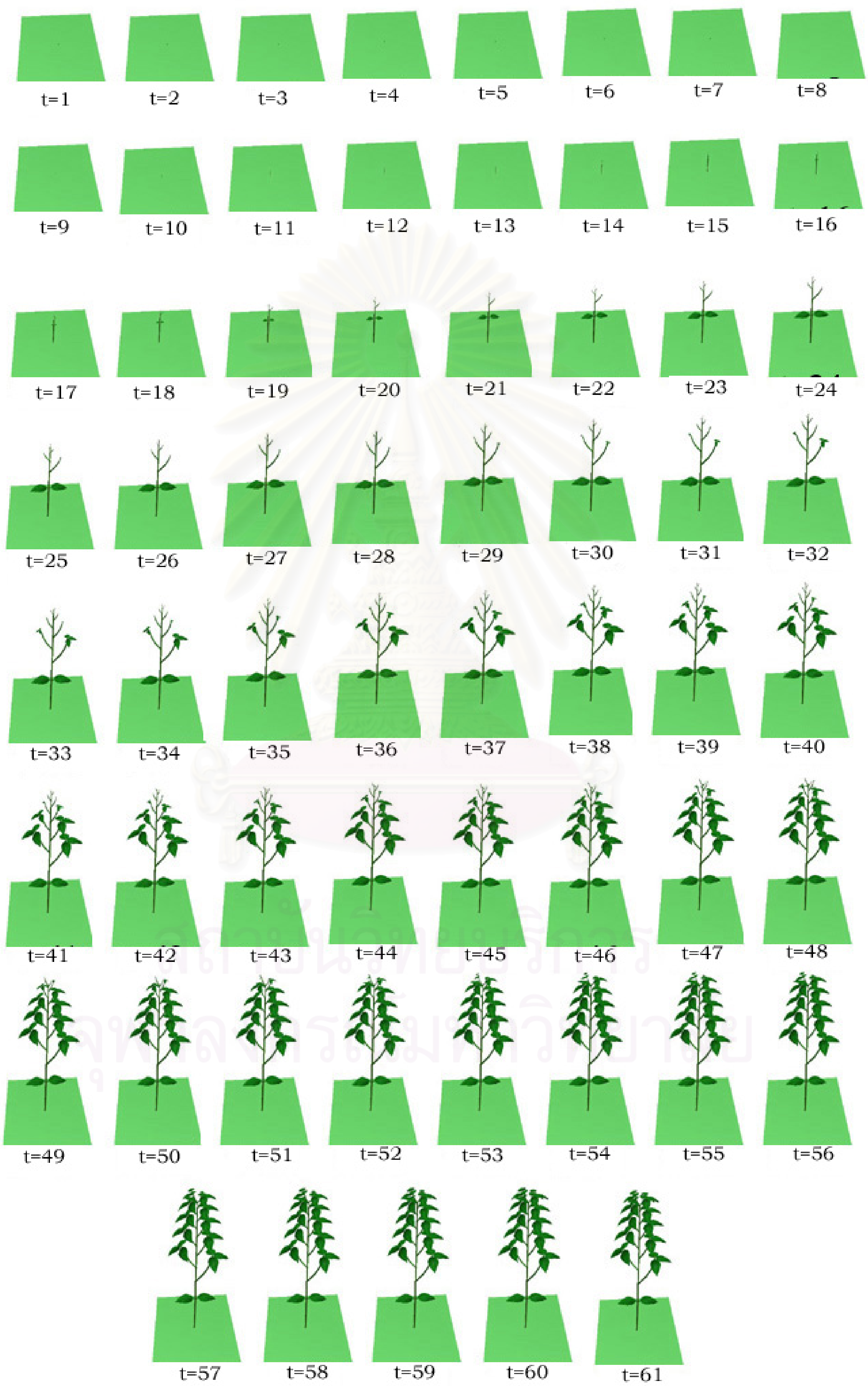


Figure 5.1: Simulation and visualization of Soybean shoots expansion over 61 days.

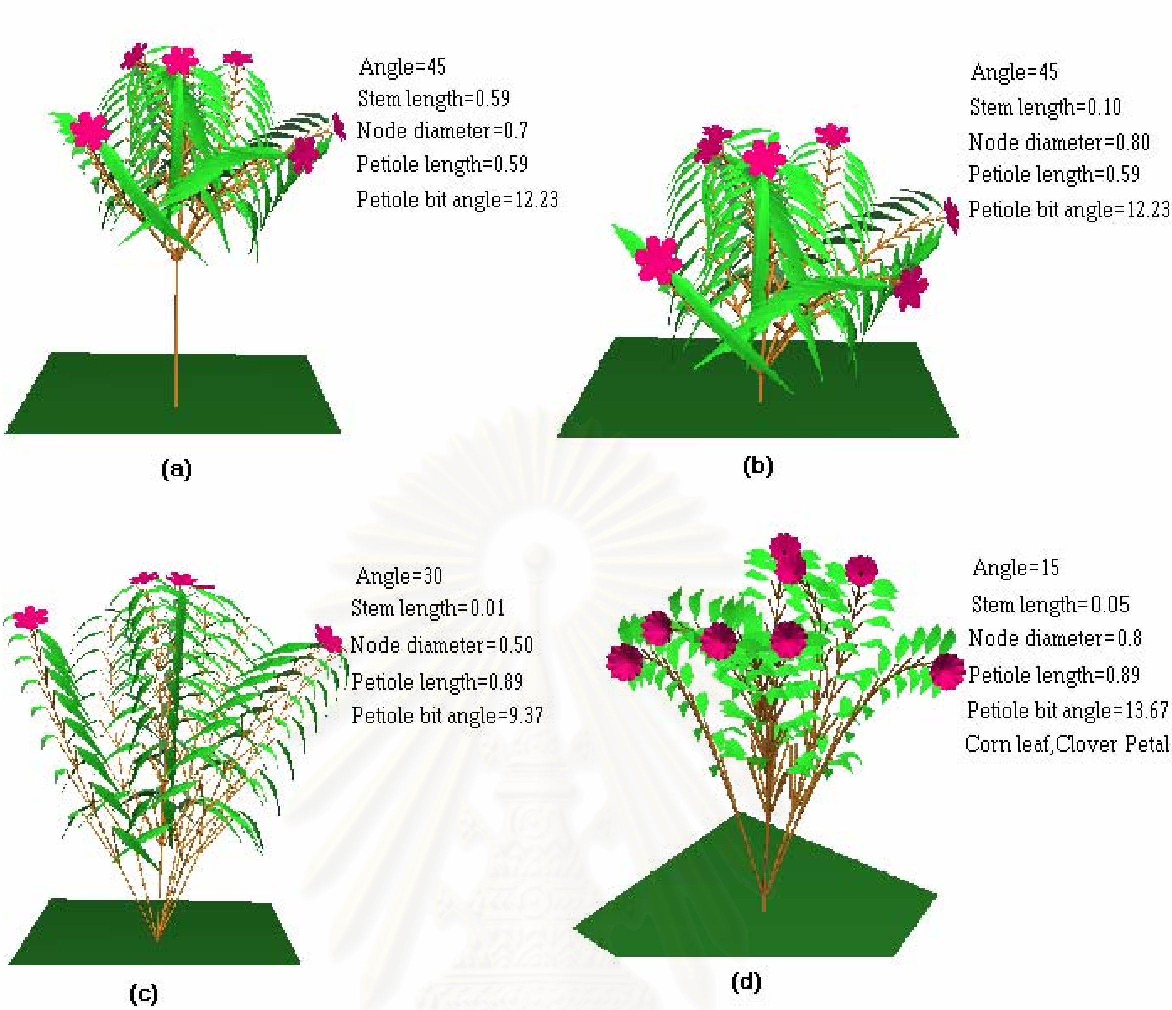


Figure 5.2: Different parameters of same topology of L-system.

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.

The PlantVR prototype can be used to generate any plant topology based on the Bracketed L-systems. The other plants example is given below. The spiral plant prototype is

```
SpiralPlant{
    Iterations=1
    Angle=45
    Diameter=2
    Axiom=IA
    A=I[^IP]I[^^IP]I[^^^IP]I[^^^^IP]I[^^^^^IP]I[^^^^^^IP]I[^^^^^^^IP]I
        [^^^^^^^^IP]IIAII[IIIIF]
    P=I[/IL]
}
```

The average data over 61 days are

Data

{

0	0.08993105
1	0.11488685
2	0.146561154
3	0.186634437
4	0.237129366
5	0.300433251
6	0.3792909
7	0.476747324
8	0.59601461
9	0.74023599
10	0.912127619
11	1.113500694
12	1.344707107
13	1.604106504
14	1.887703344
15	2.189117496
16	2.5
17	2.810882504
18	3.112296656
19	3.395893496
20	3.655292893
21	3.886499306
22	4.087872381
23	4.25976401
24	4.40398539
25	4.523252676
26	4.6207091
27	4.699566749
28	4.762870634
29	4.813365563
30	4.853438846
31	4.88511315
32	4.91006895
33	4.929681865
34	4.945065287
35	4.957112573
36	4.966535745
37	4.973899372
38	4.979649311
39	4.984136586
40	4.987636884
41	4.990366327
42	4.992494089
43	4.994152449
44	4.995444744
45	4.996451648
46	4.997236107

47 4.997847215
48 4.998323249
49 4.998694048
50 4.998982865
51 4.999207819
52 4.999383027
53 4.999519488
54 4.999625769
55 4.999708544
56 4.999773011
57 4.999823219
58 4.999862322
59 4.999892775
60 4.999916493
61 4.999934964
}

The growth function parameters are given below.

Bottom=0.0899, Top=4.9999, Slope=0.25, Tmid=16
For all component

The component parameters are given below.

```
Component
{
	Leaf Library=Leaf,Bamboo, size=24
	LeafScale  x=0.60, y=0.8, 0.9
	LeafAngle  x=61, y=0, z=0
	Flower Library=Petal, Canterbury bells, size=24, No.Petals=20,
	LeafScale  x=0.60, y=0.8, 0.9
	LeafAngle  x=61, y=0, z=0
}
```

The parameters are given below.

```
Parameter
{
	Stem length=0.35
	Node Diameter=2.0
	Node Birth Rate=4.9
	Petiole length=0.12
	Petiole BitAngle=2.16
	Internode Reduce=0.9
	Petiole Reduce=0.9
	Leaf Reduce=0.95
	Flower Reduce=0.95
	Short Internode Ratio=0.1
	Short Internode Diameter=0.8
	Short Petiole Ratio=0.1
	Short Petiole Diameter=0.8
}
```

The visualized image of above prototype, data, and parameter are shown in Figure 5.3. The Figure 5.4 shows the spiral plant with new leaf and flower shape. The growth data are approximated by the same data for every component.

Figure 5.3: The spiral plant.

Figure 5.4: The spiral plant with the new different leaves and flowers.

The simple tree prototype is given below.

```
SimpleTree{
    Iterations=2
    Angle=25
    Diameter=2
    Axiom=I[-1][+1][/1][\1]
   1=I[-1][+1][/1][\1][&/IL][&///IL]iiF
   endrule
   1=I[-IL][+IL]IiF
}
```

The visualized image of a simple tree example is shown in Figure 5.5.

Figure 5.5: The simple tree.

The example plant is applied from Soybean prototype with one iteration. The L-system code is given below. The visualized image is shown in Figure 5.6.

```
AppliedSoybean{
    Iterations=1
    Angle=45
    Diameter=2
    Axiom=I[-P]I[+B]A
    A=I[/P]IL[\B]AF
    P=III[\IL][/IL][-IL]IF
    B=II[\IL][/IL][+IL]IF
    Endrule
    F=I[-IF][+IF][/IF][\IF]IF
    F=I[-IF][+IF][/IF][\IF]IF
    B=IL
}
```

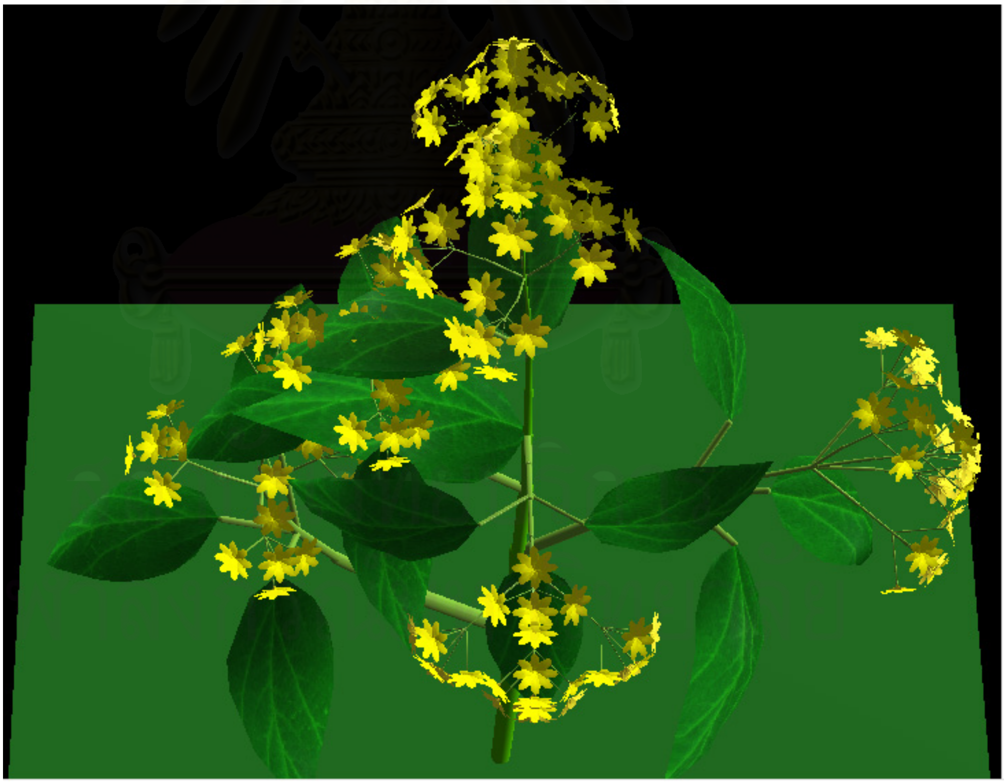


Figure 5.6: The applied soybean plant.

Chapter 6

Concluding Remarks

A prototype program called PlantVR has presented for continuous development of plant models by parametric functional symbols based on bracketed L-systems using soybean model as a case study. The proposed method consists of an order of steps:

- defining a qualitative model as L-systems,
- measuring key characteristics collected from actual plants,
- converting raw data to growth function based on sigmoidal curve,
- defining a quantitative model,
- visualizing the quantitative model, and
- evaluating the model and parameter adjustment.

The measurement of plant structure is time-consuming. The data of soybean was collected manually using rulers and protractors. This prototype can be used to generate a realistic model of any plant whose life cycle is similar to soybean.

One suggestion of this research is to add environmental parameters that need in plant growing to control the development of plant. The software can be used to generate an entire growth cycle for any plant including the reproductive state. To enhance the appearance of the plant, texture image can be applied to every component such as leaf, internode, flower, petiole, and fruit.

The advantages of the thesis are a very easy understanding of the L-systems coding and the smoother of animation that compared to the previous work. The disadvantages of the thesis is the prototype has not included the natural environment for plant growing, but this disadvantage will be extend for a future plan of this research.

The further works also to improve the component of plant, and the underground part and reproductive state will be studied.

To compare the previous work, the visualized images of L-studio software and PlantVR software are given in Figure 6.1, Figure 6.2, Figure 6.3, and Figure 6.4.

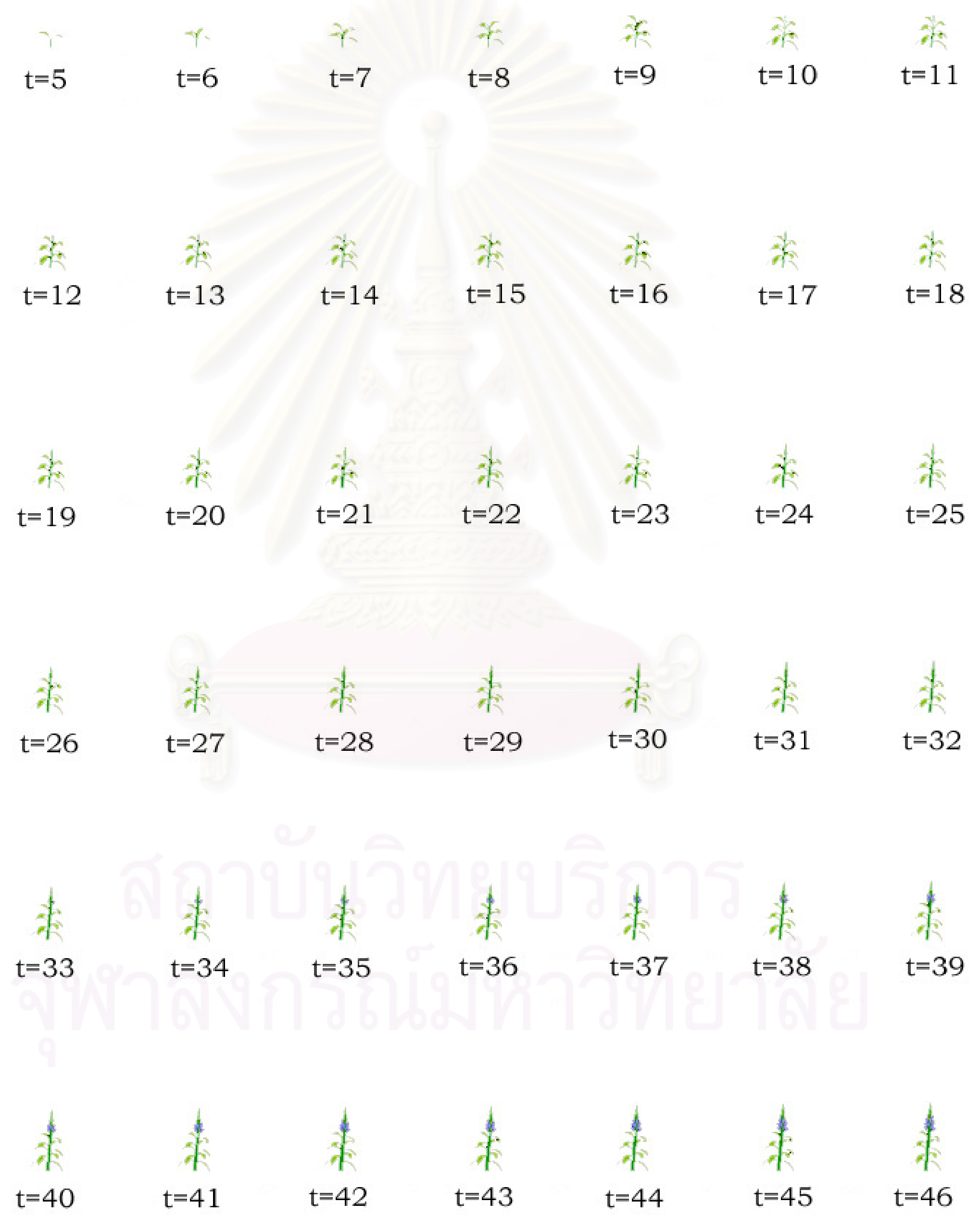


Figure 6.1: The plant growth using L-studio software at time t=5 to t=46.

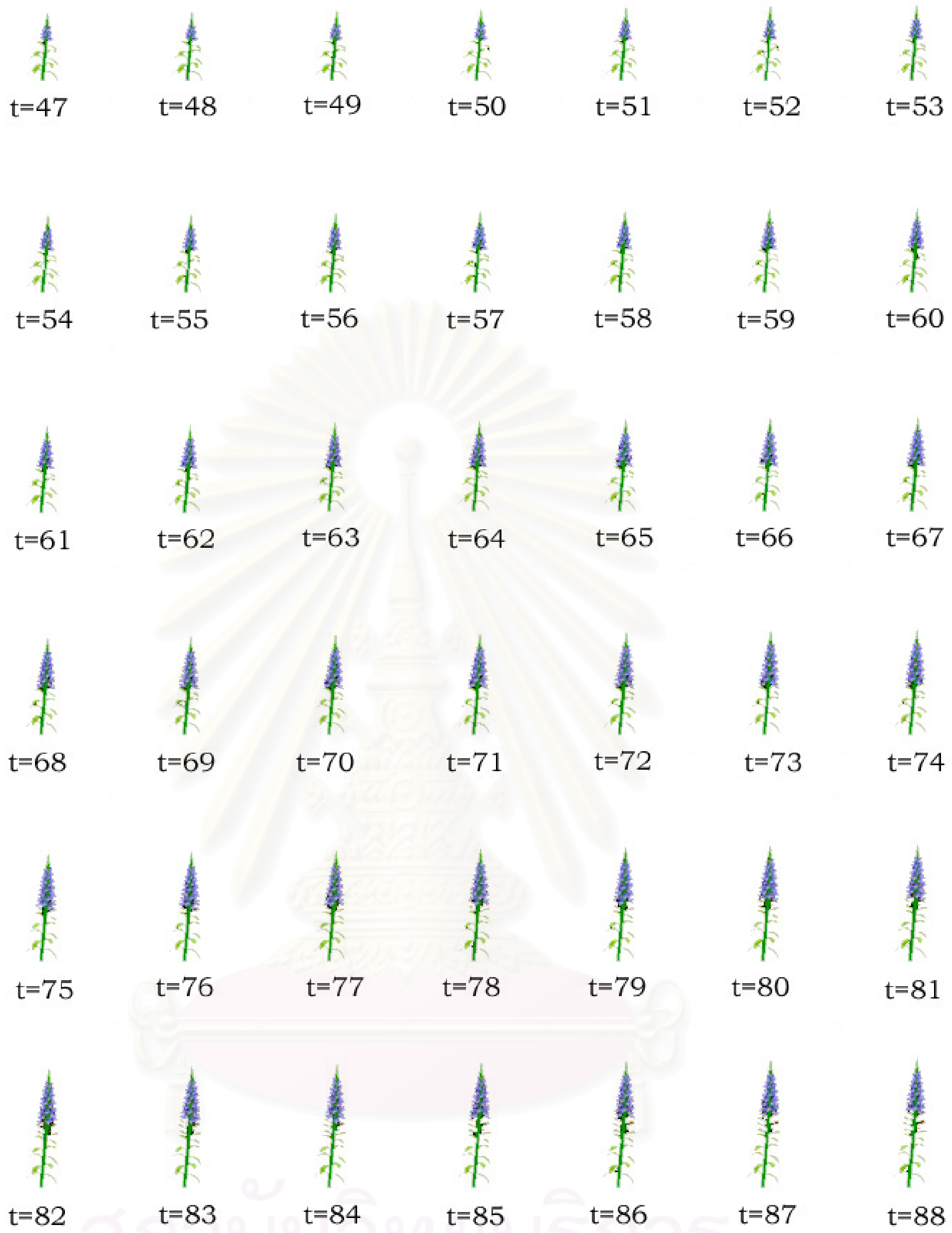


Figure 6.2: The plant growth using L-studio software at time t=47 to t=88.

Figure 6.3: The plant growth using PlantVR software at time t=5 to t=46 with growth rate 4.0.

Figure 6.4: The plant growth using PlantVR software at time t=47 to t=88 with growth rate 4.0.

To compare the L-systems codes with the similar visualized image, the L-studio software and the PlantVR software are given below.

The L-systems of L-Studio Sotfware

```
/* #define STEPS 90 */
#define I0 30
#define S0 150
#define K0 70
#define KANG1 10
#define KANG2 20
#define SRATE 1.01
#define LRATE 1.05
#define BANG0 0
#define BANGI 1.0
#define BANGL 10.0
#define T0 5 /* the number of leaf pairs */
#define T1 20 /* the number of buds */
#define T2 40 /* the number of open flowers */
#define T3 25 /* the number of fruits */
Lsystem: 1
derivation length: T0+T1+T2+T3+4
Axiom: [-(5)/(35)#(2),(47)F(I0*10)A(0)]
/* Produce the vegetative part - decussate phyllotaxis */
A(t) : t [-(BANG0,t)!(47)~l(0.5,t)]/(180) [-(BANG0,t)!(47)~l(0.5,t)]/(95)F(I0*6)!A
      (t+1)
/* Produce the inflorescence - spiral phyllotaxis */
A(t) : t >= T0 --> [&(BANG0+15)F(S0/4)&(15)F(S0/8)] [!(2)&(BANG0)G(S0)X
(0)~b(0.5)]/(137.5)F(I0)!A(t+1)
b(s) --> b(s*SRATE)
X(t) : t X(t+1)
X(t) : t==T1 --> [C(0),(127)K/(72),K/(72),K/(72),K/(72),K!(2),(122)G(120)]%
K --> [!(1)&(KANG1){-(KANG2)F(K0)+(KANG2)F(K0)+(KANG2)F(K0)-
      (KANG2)| -(KANG2)F(K0)+(KANG2)F(K0)+(KANG2)F(K0)}]
```

```
C(t) : t C(t+1)
C(t) : t==T2 --> D%
D --> ~f(0.3),(121)F(80)&(10)F(60)&(10)F(40)
F(s) --> F(s*SRATE)
f(s) --> f(s*SRATE)
G(s) --> G(s*SRATE)
l(s,t) : t<1.2 && l(s*LRATE,t+1)
&(a) : a<135 --> &(a+BANGI)
-(a,t) : a<90 && t -(a+BANGL,t+1)
#(r) : r<6 --> #(r+0.1)
endlsystem
```

The L-systems of PlantVR Sotfware

```
ComparePlant1{
        Iterations=9
        Angle=45
        Diameter=1.5
        Axiom=iiii[-iL][+iL]BA
        A=I[-IF][+IF][/IF][\IF]AK
        ENDRULE
        A=I[-IF][+IF][/IF][\IF]K
        K=IF
        B= I[/iL][\iL]I[-iL][+iL]I[/iL][\iL]i
}
```

The L-systems codes of the L-studio software and PlantVR software are very different in the meaning of definition and the axiom string. The L-studio software uses an initial string to interpret the virtual plant which the graphics are performed at each time step t in Figure 6.5(a). On the other hand, the PlantVR software iterates an initial string to perform the final L-system string. The final L-system string which controlled by growth functions is interpreted to the virtual plant in each time step t. The axiom string of “ComparePlant1” prototype is shown in graphic form as Figure 6.5(b).

The L-systems of PlantVR software are shorter than the L-studio software. The visualized images of two methods are similar, but the initial images are different. The initial image of L-studio software depends on an axiom which design at an observation position while the initial image of PlantVR depends on an actual plant development by mean of the growth function.

Another axiom of the L-systems code "ComparePlant2" of PlantVR software is shown in Figure 6.5(c). Notice that, the first frame of the animation by the PlantVR software is not similar to the axiom "ComparePlant1", but the final L-system string of "ComparePlant1" and "ComparePlant2" have the same prototype. The growth rate that used to control the final string of PlantVR can be changed, the visualized images of different growth rates are shown in Figure 6.3-6.4 and Figure 6.6-6.7. Figure 6.7 shows that the growth function is stabled at time step t = 57 which means the image of life cycle of any plant is controlled by the growth rate.

```
ComparePlant2{
        Iterations=9
        Angle=45
        Diameter=1.5
        Axiom=iiii[-iL][+iL]I[/iL][\iL]BA
        A=I[-IF][+IF][/IF][\IF]AK
        ENDRULE
        A=I[-IF][+IF][/IF][\IF]K
        K=IF
        B=I[-iL][+iL]I[/iL][\iL]i
}
```


(a) (b) (c)

Figure 6.5: The visualized image of axiom
(a) L-studio software,
(b) PlantVR software with "ComparePlant1" prototype,
(c) PlantVR software with "ComparePlant2" prototype.

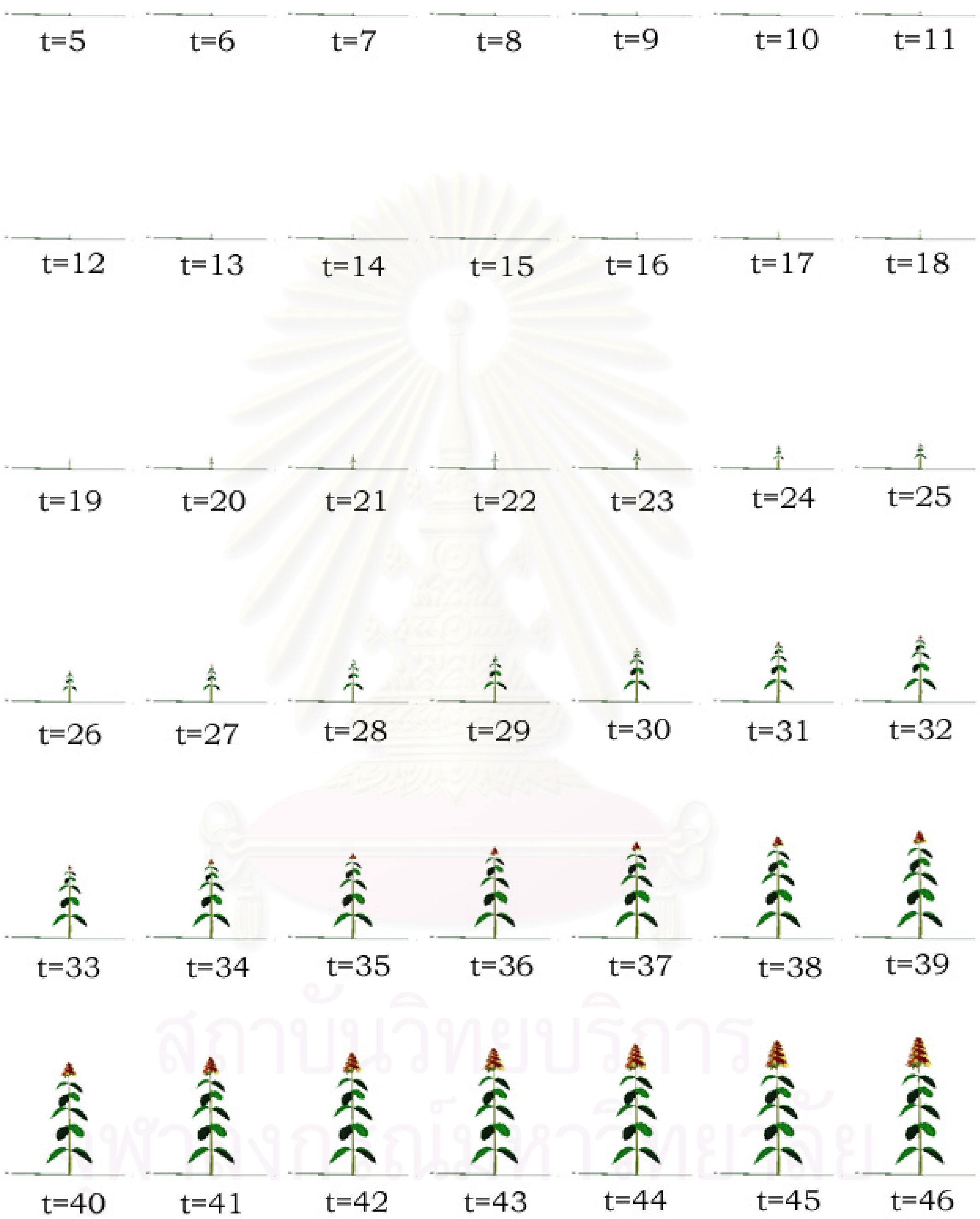


Figure 6.6: The plant growth using PlantVR software at time t=5 to t=46 with growth rate 1.70.

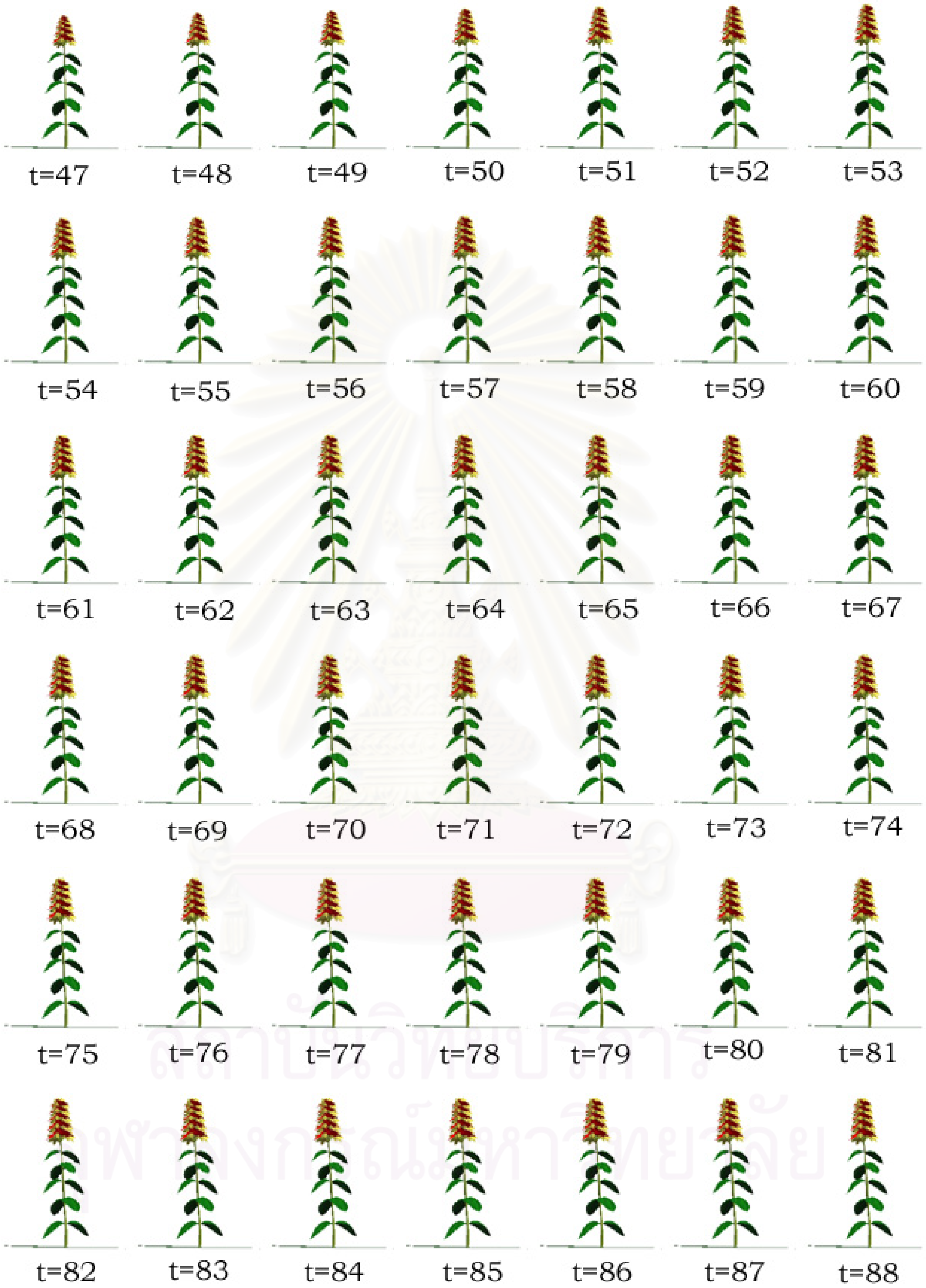


Figure 6.7: The plant growth using PlantVR software at time t=47 to t=88 with growth rate 1.70.

References

1. Amin, A. “Voxel Space Automata”, website:
   http://www.tc.cornell.edu/Visuation/contrib/cs490-94to95/Alerk/growth.html
2. Chen, H. W. “L-system Plant Geometry Generator”, website:
   http://www.tc.cornell.edu/Visualization/contrib/cs490-94to95/hwchen/
3. Deussen, O., Hanrahan, P., Linterman, B., Mech, R., Pharr, M., and Prusinkiewicz, P. Realistic modeling and rendering of plant ecosystems, SIGGRAPH’98 (July 1998) : 275-286.
4. Diaz-Ambrona, C. H., Tarquis, A. M., and Minguez, M. I. 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) : 1-13.
5. Greenworks, “Home page of the xfrog modelling software.”, website : http://www.greenworks.de.
6. Hammel, M. S. “BMV Publications”, website:
   http://www.cpsc.ucalgary.ca/projects/bmv/papers/index.html
7. Hammel, M. S. “L-systems Software”, website:
   http://www.cpsc.ucalgary.ca/projects/bmv/software.html
8. Hammel, M. S., and Prusinkiewicz, P. Visualization of developmental processes by extrusion in space-time. Proceedings of Graphics Interface ’96 (May 1996): 246-258.
9. Hammel, M. S., Prusinkiewicz, P., and Wyvil, B. Modelling Compound Leaves Usin Implicit Countours. Proceedings of CG International’ 92 (June 1992) : 119-212.
10. Hanan, J. “Centre for Plant Architecture Informatics”, website:
    http://www.cpai.uq.edu.au/
11. Hanan, J. Virtual plants---Integrating architectural and physiological plant models. Proceedings of ModSim 95 vol. 1 (1995) : 44-50.

12. Hanan, J. S., and Room, P. M. Practical aspects of virtual plant research. Plant to ecosystems: Advances in Computational Life Sciences vol. 1 (1997) : 28-44.

13. Horling, B. “Implementation of a context-sensitive Lindenmayer-system modeler” , website: http://www2.trincoll.edu/~bhorling/lsystems/

14. Kurth, W. Morphological models of plant growth: Possibilities and ecological relevance. Ecological Modelling vol. 75 (1994) : 299-308.

15. Kurth, W. “Plant modelling group”, webites: http://www.uni-forst.gwdg.de/~wkurth

16. Lapre, L. “L-systems and Lparser”, website: http://www.xs4all.nl/~ljlapre/

17. Lin, T. “Animation of L-system based 3-D plant Growing in Java”, website : http://www.cs.umbc.edu/~ebert/693/TLin/

18. Lindenmayer, A. Mathematical models for cellular interaction in development, Parts I and II. Journal of Theoretical Biology vol. 18 (1968) : 280-315.

19. Lintermann, B., and Deussen, O. A Modelling Method and User Interface for Creating Plants. The Eurographics Association vol. 17, no. 1 (March 1998) : 73-82.

20. Lintermann, B., and Deussen, O. Interactive Modeling of Plants. IEEE Computer Graphics and Applications vol. 19 no. 1 (January 1999) : 56-65.

21. Mech, R., and Prusinkiewicz, P. Visual models of plants interacting with their environment. Proceeings in Computer Graphics (SIGGRAPH’96) (August 1996) : 397-410.

22. Michalewicz, M. T. Plants to ecosystems Advances in Computational Life Sciences. Melbourne CSIRO, 1997.

23. Ochoa, G. “An Introduction to Lindenmayer Systems”, website: http://www.cogs.susx.ac.uk/users/gabro/lsys/lsys.html

24. Prusinkiewicz, P., and Hammel, M. S. Escape-time Visualization Method for Language-restricted Iterated Function Systems. Proceeding of Graphics Interface '92 (May 1992) : 213-223.

25. Prusinkiewicz, P., Hammel, M. S., Hanan, J., and Mech, R. Handbook of formal language: Visual model of plant development Springer-Verlag, 1996.

26. Prusinkiewicz, P., and Kari, L. Subapical bracketed L-systems. Lecture Notes in Computer Science vol. 1073 (1996) : 550-564.

27. Prusinkiewicz, P., and Lindenmayer, A. The Algorithmic Beauty of Plants New York: Springer-Verlag, 1990.

28. Prusinkiewicz, P., Hammel, M. S., and Mjolsness, E. Animation of plant development. Kajiya, J. T. Proceedings of Computer graphics SIGGRAPH'93 (August 1993) : 351-360.

29. Prusinkiewicz, P., James, M., and Mech, R. Synthetic topiary. Proceedings of SIGGRAPH' 94 (July 1994) : 351-358.

30. Prusinkiewicz, P., Remphrey, W., Davidson, C., and Hammel, M. Modeling the architecture of expanding *Fraxinus pennsylvanica* shoots using L-systems. Canadian Journal of Botany vol. 72 (1994) : 701-714.

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

32. Room, P. M., Hanan, J. S., and Prusinkiewicz, P. Virtual plants: New perspectives for ecologists, pathologists, and agricultural scientists. Trends in Plant Science vol. 1, no. 1(1996) : 33-38.

33. Room, P. M., Maillette, L., and Hanan, J. Module and metamer dynamics and virtual plants. Advances in Ecological Research vol. 25 (1994) :105-157.

34. Samal, A., Peterson, B., and Holliday, D. J. Recognizing Plants Using Stochastic L-systems. IEEE International Conference vol. 1 (1994) : 183-187.

35. Smith, A.R. Plants, fractals, and formal languages, ACM SIGGRAPH vol. 18, no. 3 (1984) : 1-10.

Appendices

Appendix A

Finding Normals for Analytic Surface

Analytic surfaces are smooth, differentiable surfaces that are described by a mathematical equation (or set of equations). In many cases, the easiest surfaces to find normals for are analytic surfaces for which you have an explicit definition in the following form:

V(s,t)=[X(s,t) Y(s,t) Z(s,t)]

where *s* and *t* are constrained to be in some domain, and X, Y, and Z are differentiable functions of two variables. To calculate the normal, find

$$\frac{\partial V}{\partial s} \quad and \quad \frac{\partial V}{\partial t}$$

which are vectors tangent to the surface in the *s* and *t* directions. The cross product

$$\frac{\partial V}{\partial s} \times \frac{\partial V}{\partial t}$$

is perpendicular to both and, hence, to the surface. The following shows how to calculate the cross product of two vectors. (Watch out for the degenerate cases where the cross product has zero length).

$$\lfloor v_x \ v_y \ v_z \rfloor \times \lfloor w_x \ w_y \ w_z \rfloor = \lfloor (v_y w_z - w_y \ v_z)\,(w_x v_z - v_x \ w_z)\,(v_x w_y - w_x \ v_y) \rfloor$$

The resulting vector should be normalized. To normalize a vector [x y z], calculate its length

$$Length = \sqrt{x^2 + y^2 + z^2}$$

and divide each component of the vector by the length.

As an example of these calculations, consider the analytic surface

$V(s, t)=[\ s^2\ t^3\ 3\text{-}st]$

From this

$$\frac{\partial V}{\partial s} = [2s\ 0 - t], \quad \frac{\partial V}{\partial t} = [0\ 3t^2\ -s], \ and\ \frac{\partial V}{\partial s} \times \frac{\partial V}{\partial t} = [-3t^3\ \ 2s^2\ \ 6st^2],$$

So, for example, when s=1 and t=2, the corresponding point on the surface is (1,8,1), and the vector (-24,2,24) is perpendicular to the surface at that point. The length of this vector is 34, so the unit normal vector is (-24/34, 2/34, 24/34) = (-0.70588, 0.058823, 0.70588).

For analytic surfaces that are described implicitly, as F(x, y, z) = 0, the problem is harder. In some cases, on of the variables can be solved, z = G(x, y), and put it in the explicit form given previously:

$V(s, t) = [s \quad t \ \ G(s, t)]$

Then continue as described earlier.

If you can't get the surface equation in an explicit form, you might be able to make use of the fact that the normal vector is given by the gradient

$$\nabla F = \left[\frac{\partial F}{\partial x}\ \frac{\partial F}{\partial y}\ \frac{\partial F}{\partial z} \right]$$

evaluated at a particular point (*x, y, z*). Calculating the gradient might be easy, but finding a point that lies on the surface can be difficult. As an example of an implicitly defined analytic function, consider the equation of a sphere of radius 1 centered at the origin:

$x^2 + y^2 + z^2 – 1 = 0$

This means that

$$F(x, y, z) = x^2 + y^2 + z^2 - 1$$

$$z = \pm\sqrt{1 - x^2 - y^2}$$

which can be solved for z to yield

Thus, normals can be calculated from the explicit from

$$V(s,t) = \left[s \quad t \quad \sqrt{1 - s^2 - t^2} \right]$$

as described previously.

If you could not solve for z, you could have used the gradient

$$\nabla F = [2x\ 2y\ 2z]$$

as long as you could find a point on the surface. In this case, it's not so hard to find a point --- for example, (2/3, 1/3, 2/3) lies on the surface. Using the gradient, the normal at this point is (4/3, 2/3, 4/3). The unit-length normal is (2/3, 1/3, 2/3), which is the same as the point on the surface, as expected.

Finding Normals from Polygonal Data

As mentioned previously, you often want to find normals for surfaces that are described with polygonal data such that the surfaces appear smooth rather that faceted. In most cases, the easiest way for you to do this is to calculate the normal vectors for each of the polygonal facets and then to average the normals for neighboring facets. Use the averaged normal for the vertex that the neighboring facets have in common. Figure A-1 shows a surface and its polygonal approximation.

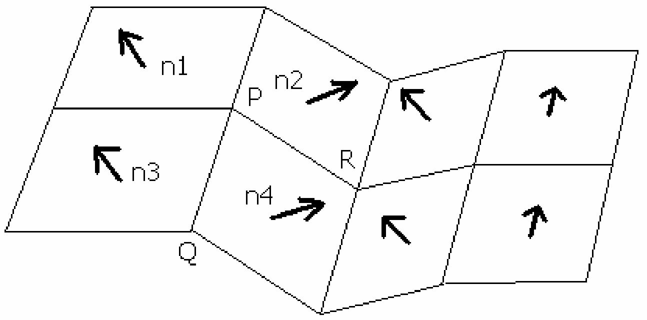


Figure A-1: Averaging and normal vecters.

To find the normal for a flat polygon, take any three vertices v1, v2, and v3 of the polygon that do not lie in a straight line. The cross product

$[v_1 - v_2] \times [v_2 - v_3]$

is perpendicular to the polygon. (Typically, you want to normalize the resulting vector.) Then you need to average the normals for adjoining facets to avoid giving to much weight to one of them. For instance, in the example shown in Figure A-1, if n_1, n_2, and n_3 are the normals for three polygons meeting a t point p, calculated $n_1+n_2+n_3$ and the normalize it. (you can get a better average if you weight the normals by the size of the angles at the shared intersection.) The resulting vector can be used as the normal for point P.

Homogeneous Coordinates

OpenGL commands usually deal with two-and three-dimensional vertices, but in fact all are treated internally as three-dimensional homogeneous vertices comprising four coordinates. Every column vector $(x, y, z, w)^T$ represents a homogeneous vertex if at least on of its elements is nonzero. If the real number *a* is nonzero, the $(x, y, z, w)^T$ and $(ax, ay, az, aw)^T$ represent the same homogeneous vertex. (This is just like fractions: $x/y = (ax)/(ay)$.) A three-dimensional euclidean space point $(x, y, z)^T$ becomes the homogeneous vertex with coordinates $(x, y, z, 1)^T$, and the two-dimensional euclidean point $(x, y)^T$ becomes $(x, y, 0.0, 1.0)^T$.

As long as *w* is nonzero, the homogeneous vertex $(x, y, z, w)^T$ corresponds to the three-dimensional point $(x/w, y/w, z/w)^T$. Fi w = 0.0, it corresponds to no euclidean point, but rather to some idealized “point at infinity”. To understand this point at infinity, consider the point (1, 2, 0, 0), and note that the sequence of points (1, 2, 0, 1), (1, 2, 0, 0.01), and (1, 2.0, 0.0, 0.0001), corresponds to the euclidean points (1, 2), (100, 200) and (10000, 20000). This sequence represents points rapidly moving toward infinity along the line 2x = y. Thus, you can think of (1, 2, 0, 0) as the point at infinity in the direction of that line.

Note: OpenGL might not handle homogeneous clip coordinates with x<0 correctly. To be sure that your code is portable to all OpenGL systems, use only nonnegative *w* values.

Transforming Vertices

Vertex transformations (such as rotations, translations, scaling, and shearing) and projections (such as perspective and orthographic) can all be represented by applying an appropriate 4x4 matrix to the coordinates representing the vertex. If *v* represents a homogeneous vertex and **M** is a 4x4 transformation matrix, the **Mv** is the image of **v** under the transformation by **M**. In computer-graphics applications, the transformations used are usually nonsingular—in other words, the matrix **M** can be inverted. This is not required, but some problems arise with nonsingular transformations.

After transformation, all transformed vertices are clipped so that x, y, and z are in the range [-*w*, *w*] (assuming *w>0*). Note that this range corresponds in euclidean space to [-1.0, 1.0].

Transforming Normals

Normal vectors are not transformed in the same way as vertices or position vectors. Mathematically, it is better to think of normal vectors not as vectors, but as planes perpendicular to those vectors. Then, the transformation rules for normal vectors are described by the transformation rules for perpendicular planes.

A homogeneous plane is denoted by the row vector (a, b, c, d), where at least one of a, b, c, or d is nonzero. If q is a nonzero real number, then (a, b, c, d) and (qa, qb, qc, qd) represent the same plane. A point $(x, y, z, w)^T$ is not the plane (a, b, c, d) if ax+by+cz+dw=0. (If w=1, this is the standard description of a euclidean plane.) In order for (a, b, c, d) to represent a euclidean plane, at least one of a, b, or c must be nonzero. If they are all zero, then (0, 0, 0, d) represents the "plane at infinity," which contains all the "points at infinity."

If **p** is a homogeneous plane and **v** is an homogeneous vertex, then the statement "**v** lies on plane **p**" is written mathematically as **pv=0**, where **pv** is normal matrix multiplication. If **M** is a nonsingular vertex transformation (that is, a 4x4 matrix that has an inverse $\mathbf{M^{-1}}$), then **pv=0** is equivalent to $\mathbf{pM^{-1}Mv=0}$, so **Mv** lies on the plane $\mathbf{pM^{-1}}$. Thus, $\mathbf{pM^{-1}}$ is the image of the plane under the vertex transformation **M.**

If you like to think of normal vectors as vectors instead of as the planes perpendicular to them, let **v** and **n** be vectors such that **v** is perpendicular to **n.** Then, $\mathbf{n^Tv=0}$, thus, for an arbitrary nonsingular transformation **M,** $\mathbf{n^TM^{-1}Mv=0}$, which means that $\mathbf{n^TM^{-1}}$ is the transpose of the transformed normal vector. Thus, the transformed normal vector is $\mathbf{(M^{-1})^Tn}$. In order words, normal vectors are transformed by the inverse transpose of the transformation that transforms points.

Transformation Matrices

Although any nonsingular matrix **M** represents a valid projective transformation, a few special matrices are particularly useful. These matrices are listed in the following subsections.

Translation

The call **glTranslate***(x, y, z) generates **T**, where

$$T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad and \quad T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

The call **glScale***(x, y, z) generates **S**, where

$$S = \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad and \quad S^{-1} = \begin{bmatrix} \frac{1}{x} & 0 & 0 & 0 \\ 0 & \frac{1}{y} & 0 & 0 \\ 0 & 0 & \frac{1}{z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notice that S^{-1} is defined only if x, y, and z are all nonzero.

Rotation

The call **glRotate***(a, x, y, z) generates R as follows:

Let $v = (x, y, z)^T$, and $u = v/||v|| = (x', y', z')^T$.

Also let

$$S = \begin{bmatrix} 0 & -z' & y' \\ z' & 0 & -x' \\ -y' & x' & 0 \end{bmatrix}$$

and $M = uu^T + (\cos a)(I - uu^T) + (\sin a)\, S$

Then

$$R = \begin{bmatrix} m & m & m & 0 \\ m & m & m & 0 \\ m & m & m & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where *m* represents elements from M, which is a 3x3 matrix.

The **R** matrix is always defined. If x = y = z =0, then **R** is the identity matrix. You can obtain the inverse of **R**, $\mathbf{R}^{-1}$, by substituting –*a* for *a*, or by transposition.

The **glRotate*()** command generates a matrix for rotation about an arbitrary axis.

The corresponding matrices are as follows:

$$glRotate*(a,1,0,0): \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos a & -\sin a & 0 \\ 0 & \sin a & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$glRotate*(a,0,1,0): \begin{bmatrix} \cos a & 0 & \sin a & 0 \\ 0 & 1 & 0 & 0 \\ -\sin a & 0 & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$glRotate*(a,0,0,1): \begin{bmatrix} \cos a & -\sin a & 0 & 0 \\ \sin a & \cos a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As before, the inverses are obtained by transposition.

Perspective Projection

The call glFrustum(*l, r, b, t, n, f*) generates **R**, where

$$R = \begin{bmatrix} \frac{2n}{r-1} & 0 & \frac{r+1}{r-1} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad and \quad R^{-1} = \begin{bmatrix} \frac{r-1}{2n} & 0 & 0 & \frac{r+1}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{-(f-n)}{2fn} & \frac{f+n}{2fn} \end{bmatrix}$$

R is defined as long as $l \neq r$, $t \neq b$, and $n \neq f$.

Orthographic Projection

The call **glOrtho(*l, r, b, t, n, f*)** generates **R**, where

$$R = \begin{bmatrix} \frac{2}{r-1} & 0 & 0 & -\frac{r+1}{r-1} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad and \quad R^{-1} = \begin{bmatrix} \frac{r-1}{2} & 0 & 0 & \frac{r+1}{2} \\ 0 & \frac{t-b}{2} & 0 & \frac{t+b}{2} \\ 0 & 0 & \frac{f-n}{-2} & \frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R is defined as long as $l \neq r$, $t \neq b$, and $n \neq f$.

Appendix B

This appendix presents the data which are collected from three soybeans experiment over 61 days. The structure of soybean is shown in Figure B-1.

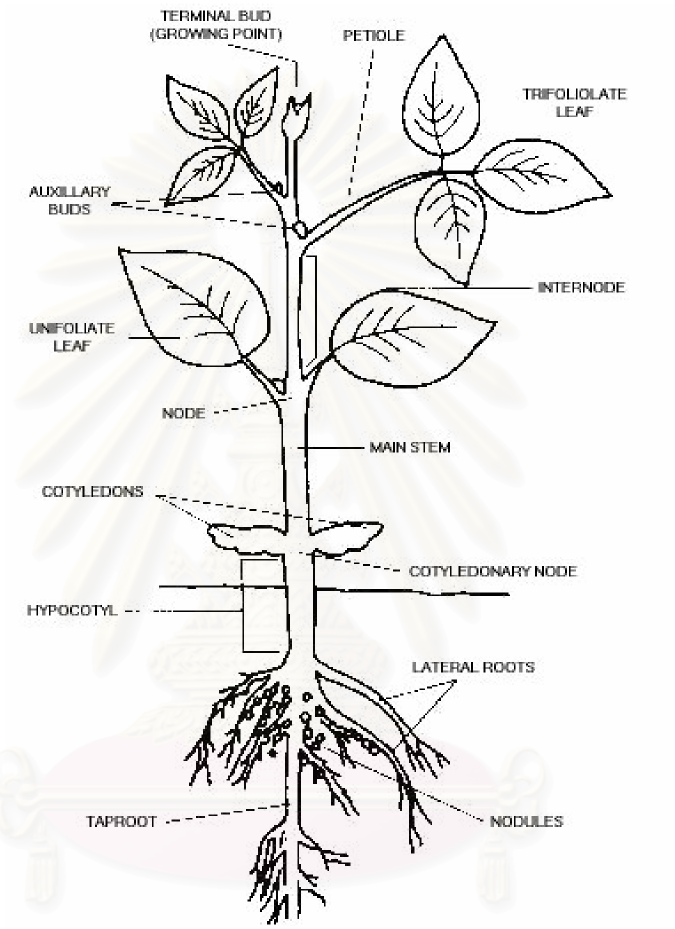


Figure B-1: Soybean physiology.

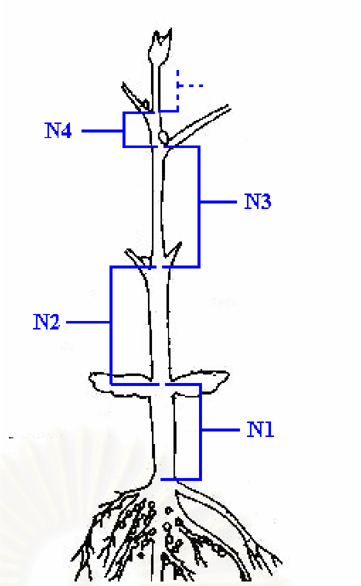


Figure B-2: Internode data collection.

Table B-1: The average over three soybeans of Internode data (N1-N10).

Date	Avg N1	Avg N2	Avg N3	Avg N4	Avg N5	Avg N6	Avg N7	Avg N8	Avg N9	Avg N10
1	2.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	5.08	0.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	6.55	1.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	7.32	1.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	7.32	2.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	7.35	4.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	7.20	5.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	7.20	5.60	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	7.20	5.85	1.40	0.25	0.00	0.00	0.00	0.00	0.00	0.00
10	7.20	5.85	1.75	0.40	0.00	0.00	0.00	0.00	0.00	0.00
11	7.10	5.85	2.15	0.60	0.00	0.00	0.00	0.00	0.00	0.00
12	7.10	5.85	2.25	0.80	0.00	0.00	0.00	0.00	0.00	0.00
13	7.00	5.80	2.45	1.30	0.40	0.00	0.00	0.00	0.00	0.00
14	7.00	5.80	2.45	1.60	0.70	0.00	0.00	0.00	0.00	0.00
15	7.00	5.75	2.25	2.10	1.15	0.00	0.00	0.00	0.00	0.00
16	7.00	5.80	2.65	2.45	1.75	0.60	0.00	0.00	0.00	0.00
17	7.00	5.80	2.65	2.60	2.40	0.70	0.15	0.00	0.00	0.00
18	7.00	5.85	2.85	2.70	2.95	1.25	0.35	0.00	0.00	0.00

19	7.00	5.75	2.80	2.60	3.15	1.90	0.50	0.00	0.00	0.00
20	7.00	5.75	2.50	2.55	3.50	2.75	0.65	0.25	0.00	0.00
21	7.00	5.75	2.50	2.65	3.55	3.50	1.05	0.45	0.00	0.00
22	7.00	6.07	2.57	2.53	3.63	3.93	1.43	0.53	0.27	0.00
23	7.00	6.13	2.63	2.53	3.50	4.00	2.33	0.77	0.30	0.00
24	7.00	6.13	2.63	2.53	3.70	4.10	3.13	1.13	0.37	0.20
25	7.00	6.13	2.63	2.53	3.70	4.87	3.80	1.63	0.53	0.30
26	7.00	6.13	2.63	2.60	3.70	4.87	4.60	2.63	0.80	0.70
27	7.00	6.13	2.63	2.60	3.70	4.87	4.77	3.77	1.17	0.57
28	7.00	6.13	2.63	2.60	3.70	4.87	4.80	4.60	1.87	0.73
29	7.00	6.13	2.63	2.60	3.70	4.90	4.83	5.33	2.90	1.03
30	7.00	6.13	2.50	2.60	3.70	4.90	4.83	5.47	3.67	1.37
32	7.00	6.17	2.60	2.57	3.57	4.67	4.77	5.47	5.80	3.53
33	7.00	6.17	2.65	2.57	3.63	4.73	4.83	5.53	5.90	5.33
34	7.00	6.17	2.70	2.63	3.70	4.87	4.90	5.53	5.90	6.53
35	7.00	6.17	2.67	3.00	3.90	4.87	4.83	5.43	6.03	6.87
36	7.00	6.17	2.57	2.60	3.63	4.80	4.90	5.40	5.77	6.77
37	7.00	6.17	2.57	2.60	3.63	4.80	4.90	5.43	5.80	6.83
38	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.87	6.87
40	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.80	6.70
41	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.77	6.83
42	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.80	6.80
43	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.80	6.80
44	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.73	6.83
45	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.73	6.83
46	7.00	6.17	2.57	2.60	3.63	4.90	4.67	5.40	5.73	6.87
49	7.00	6.17	2.57	2.67	3.67	4.87	4.77	5.43	5.80	6.77
51	7.00	6.17	2.57	2.67	3.67	4.87	4.77	5.43	5.80	6.77
53	7.00	6.17	2.50	2.53	3.57	4.77	4.73	5.40	5.73	6.87
55	7.00	6.17	2.50	2.53	3.57	4.77	4.73	5.40	5.73	6.87
58	7.00	6.17	2.50	2.53	3.57	4.77	4.73	5.40	5.73	6.87
60	7.00	6.17	2.50	2.53	3.57	4.77	4.73	5.40	5.73	6.87

Table B-2: The average over three soybeans of Internode data (N11-N20).

Date	Avg N11	Avg N12	Avg N13	Avg N14	Avg N15	Avg N16	Avg N17	Avg N18	Avg N19	Avg N20
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.27	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.30	0.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.43	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.53	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

32	1.13	0.47	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	2.57	0.70	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	3.13	0.87	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	3.37	0.97	0.40	0.20	0.00	0.00	0.00	0.00	0.00	0.00
36	5.53	1.93	0.63	0.27	0.03	0.00	0.00	0.00	0.00	0.00
37	5.90	2.23	0.70	0.30	0.00	0.00	0.00	0.00	0.00	0.00
38	6.77	4.33	1.10	0.40	0.25	0.00	0.00	0.00	0.00	0.00
40	6.87	6.47	2.83	0.77	0.35	0.13	0.00	0.00	0.00	0.00
41	6.83	6.67	4.30	1.27	0.43	0.23	0.07	0.00	0.00	0.00
42	6.87	6.73	5.50	2.00	0.63	0.23	0.12	0.03	0.00	0.00
43	6.87	6.73	5.50	2.00	0.63	0.23	0.12	0.03	0.00	0.00
44	6.77	6.80	6.23	3.80	1.03	0.33	0.17	0.03	0.00	0.00
45	6.87	6.77	6.27	4.30	1.20	0.30	0.17	0.03	0.03	0.00
46	6.87	6.67	6.27	4.63	1.43	0.50	0.23	0.13	0.03	0.00
49	6.83	6.70	6.23	5.17	2.67	0.77	0.30	0.20	0.07	0.00
51	6.83	6.70	6.23	5.23	3.17	1.20	0.37	0.23	0.07	0.03
53	6.87	6.73	6.23	5.13	3.17	1.47	0.60	0.27	0.10	0.03
55	6.93	6.73	6.20	5.17	3.20	1.53	0.67	0.27	0.10	0.03
58	6.90	6.73	6.20	5.13	3.20	1.67	0.67	0.30	0.10	0.03
60	6.90	6.73	6.27	5.17	3.23	1.73	0.67	0.30	0.10	0.03

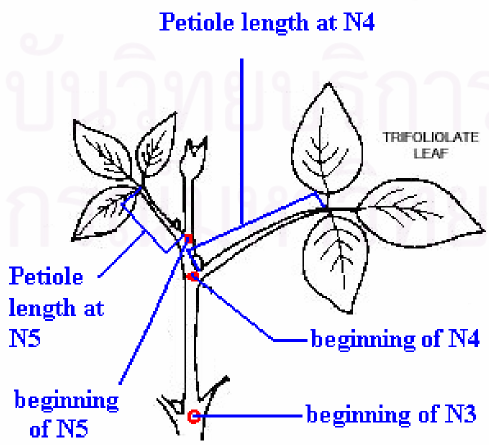


Figure B-3: Petiole data collection.

Table B-3: The average over three soybeans of petiole data (N4-N11).

Date	N4	N5	N6	N7	N8	N9	N10	N11
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	1.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	1.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	2.07	0.53	0.00	0.00	0.00	0.00	0.00	0.00
12	2.80	0.73	0.00	0.00	0.00	0.00	0.00	0.00
13	4.17	2.03	0.27	0.00	0.00	0.00	0.00	0.00
14	4.93	2.93	1.60	0.00	0.00	0.00	0.00	0.00
15	5.33	4.10	0.93	0.00	0.00	0.00	0.00	0.00
16	5.63	5.03	1.60	0.00	0.00	0.00	0.00	0.00
17	5.80	5.93	2.40	0.33	0.00	0.00	0.00	0.00
18	5.97	6.33	3.30	0.60	0.00	0.00	0.00	0.00
20	6.07	6.87	5.00	1.83	0.30	0.00	0.00	0.00
21	6.20	7.23	5.90	2.63	1.17	0.00	0.00	0.00
22	6.20	7.47	6.67	3.83	1.07	0.33	0.00	0.00
23	6.23	7.63	7.37	5.00	1.93	0.57	0.17	0.00
24	6.23	7.80	7.67	5.93	2.57	0.67	0.27	0.00
25	6.23	7.97	8.17	6.93	3.50	0.93	0.33	0.13
26	6.23	8.07	8.50	7.37	4.87	1.70	0.50	0.20
27	6.23	8.07	8.67	8.00	5.97	2.47	0.73	0.37
28	6.23	8.10	8.97	8.33	6.83	3.47	1.10	0.43
29	6.23	8.10	9.07	8.70	7.60	4.83	1.83	0.63
30	6.23	8.10	9.37	9.17	8.27	6.63	2.57	0.80
32	6.20	8.07	9.53	9.73	9.20	7.80	5.37	2.13
33	6.23	8.07	9.70	10.07	9.67	8.37	6.67	3.17

34	6.20	8.07	9.82	10.20	9.90	8.70	7.50	4.30
35	6.13	8.07	9.73	10.27	10.10	8.97	8.37	5.60
36	6.13	8.10	9.83	10.40	10.23	9.33	8.80	6.63
37	6.13	8.10	9.83	10.37	10.47	9.57	9.20	7.20
38	6.20	8.07	9.87	10.43	10.53	9.73	9.40	7.67
40	6.13	8.10	9.83	10.50	10.70	10.00	9.90	8.37
41	6.13	8.10	9.83	10.50	10.70	10.17	10.10	8.67
42	6.13	8.10	9.83	10.50	10.70	10.27	10.27	8.87
44	6.13	8.10	9.83	10.50	10.70	10.43	10.23	9.10
45	6.13	8.10	9.83	10.50	10.70	10.30	10.50	9.30
46	6.13	8.10	9.83	10.50	10.70	10.30	10.50	9.37
49	6.13	8.10	9.97	10.50	10.83	10.33	10.63	9.53
51	6.13	8.10	9.97	10.50	10.83	10.33	10.63	9.53
53	6.13	8.10	9.97	10.50	10.83	10.33	10.63	9.60
55	6.25	8.00	9.95	10.30	11.05	10.30	10.65	9.75
58	6.20	8.55	9.90	10.35	11.00	10.35	10.75	9.65
60	6.20	8.55	10.15	9.75	10.60	10.75	10.50	9.75

Table B-4: The average over three soybeans of petiole data (N12-N19).

Date	N12	N13	N14	N15	N16	N17	N18	N19
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.28	0.15	0.00	0.00	0.00	0.00	0.00	0.00
30	0.37	0.17	0.00	0.00	0.00	0.00	0.00	0.00
32	0.67	0.33	0.13	0.00	0.00	0.00	0.00	0.00
33	1.37	0.47	0.13	0.00	0.00	0.00	0.00	0.00
34	1.60	0.57	0.15	0.00	0.00	0.00	0.00	0.00
35	2.43	2.87	0.33	0.00	0.00	0.00	0.00	0.00
36	3.47	1.03	0.37	0.10	0.00	0.00	0.00	0.00
37	4.50	1.53	0.47	0.23	0.00	0.00	0.00	0.00
38	5.13	2.00	0.63	0.23	0.00	0.00	0.00	0.00
40	6.23	3.57	1.37	0.50	0.13	0.00	0.00	0.00
41	6.77	4.50	2.00	0.63	0.23	0.00	0.00	0.00
42	7.00	5.30	2.77	0.97	0.27	0.00	0.00	0.00
44	7.47	5.93	4.67	1.80	0.53	0.07	0.00	0.00
45	7.60	6.80	4.23	2.07	0.60	0.07	0.00	0.00
46	8.37	6.33	4.50	2.57	0.83	0.10	0.00	0.00
49	8.00	6.60	5.03	4.53	1.73	0.23	0.00	0.00
51	8.10	6.80	5.20	3.97	2.30	0.47	0.10	0.00
53	8.10	6.83	5.30	4.07	2.57	0.63	0.13	0.00

55	8.60	7.55	6.05	4.65	3.20	1.20	0.25	0.00
58	8.65	7.50	6.05	4.85	3.35	1.60	0.65	0.00
60	8.70	7.45	6.10	4.95	3.45	1.90	0.80	0.50

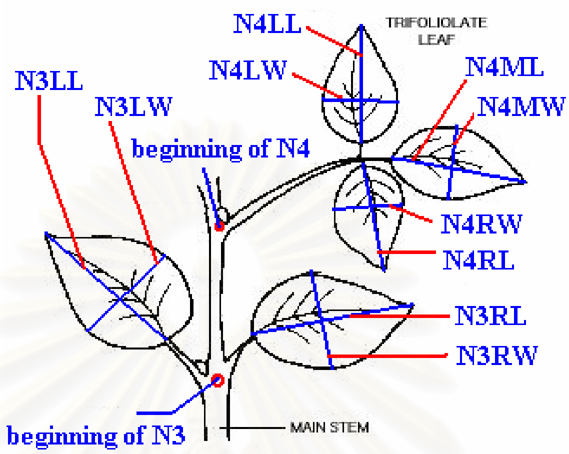


Figure B-4: Leaf data collection.

Table B-5: The average over three soybeans of leaf width data (N3-N5).

Date	N3 LW	N3 RW	N4 LW	N4 MW	N4 RW	N5 LW	N5 MW	N5 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	1.43	1.10	0.00	0.00	0.00	0.00	0.00	0.00
9	2.37	2.27	0.00	0.00	0.00	0.00	0.00	0.00
10	2.10	2.53	0.00	0.00	0.00	0.00	0.00	0.00
11	2.83	2.77	0.00	0.00	0.00	0.00	0.00	0.00
12	2.93	2.87	0.00	0.00	0.00	0.00	0.00	0.00
13	2.93	2.87	0.27	0.37	0.27	0.00	0.00	0.00
14	2.97	2.87	0.30	0.53	0.40	0.00	0.00	0.00
15	2.97	2.97	0.83	1.17	0.93	0.00	0.00	0.00
16	3.00	3.00	1.13	1.67	1.30	0.00	0.00	0.00
17	3.03	2.97	1.53	2.07	1.73	0.00	0.00	0.00
18	3.03	2.97	1.87	2.47	2.03	0.00	0.00	0.00
19	3.03	2.97	2.17	2.67	2.30	0.00	0.00	0.00
20	3.03	3.03	2.33	2.77	2.47	0.00	0.00	0.00

21	3.03	3.03	2.43	2.93	2.57	0.58	0.37	0.60
22	3.03	3.03	2.43	2.93	2.57	0.70	0.47	0.67
23	3.03	3.03	2.43	2.93	2.57	1.23	0.73	1.23
24	3.03	3.03	2.47	2.93	2.60	1.83	1.20	1.87
25	3.03	3.03	2.47	2.93	2.60	2.27	1.50	2.33
26	3.03	3.03	2.47	2.93	2.60	2.63	1.80	2.72
27	3.03	3.03	2.47	2.93	2.60	2.93	2.03	3.07
28	3.03	3.03	2.47	2.93	2.60	3.10	2.13	3.23
29	2.97	2.97	2.47	2.93	2.60	3.20	2.23	3.40
30	2.93	2.90	2.43	2.87	2.53	3.30	2.30	3.43
31	2.90	2.90	2.43	2.87	2.53	3.30	2.30	3.43
32	2.90	2.90	2.43	2.87	2.57	3.23	2.33	3.43
33	2.90	2.90	2.43	2.87	2.57	3.23	2.33	3.43
34	2.57	2.90	2.43	2.87	2.57	3.23	2.33	3.43
35	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
36	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
37	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
38	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
39	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
40	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
41	0.93	2.90	2.43	2.87	2.57	3.23	2.33	3.80
42	0.00	1.87	2.43	2.87	2.57	3.23	2.33	3.80
43	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
44	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
45	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
46	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
47	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
49	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
50	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
51	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
52	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
53	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
54	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40

55	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
56	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
57	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
58	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
60	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40
61	0.00	0.00	2.40	2.87	2.57	3.23	2.33	3.40

Table B-6: The average over three soybeans of leaf width data (N6-N8).

Date	N6 LW	N6 MW	N6 RW	N7 LW	N7 MW	N7 RW	N8 LW	N8 MW	N8 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.70	0.87	0.70	0.00	0.00	0.00	0.00	0.00	0.00
28	0.98	1.27	0.97	0.13	0.20	0.13	0.00	0.00	0.00

29	1.40	1.77	1.37	0.20	0.30	0.23	0.00	0.00	0.00
30	1.77	2.13	1.73	0.23	0.33	0.23	0.00	0.00	0.00
31	2.07	2.50	2.03	0.33	0.40	0.33	0.00	0.00	0.00
32	2.43	2.93	2.43	0.47	0.57	0.50	0.10	0.17	0.10
33	2.80	3.23	2.83	0.77	0.83	0.77	0.13	0.20	0.13
34	3.00	3.37	3.00	1.10	1.20	1.07	0.23	0.27	0.20
35	3.13	3.50	3.13	1.60	1.73	1.57	0.37	0.47	0.37
36	3.23	3.63	3.23	2.00	2.20	2.00	0.63	0.73	0.63
37	3.30	3.70	3.33	2.50	2.77	2.53	1.07	1.30	1.13
38	3.30	3.70	3.33	2.80	3.03	2.83	1.40	1.72	1.50
39	3.33	3.73	3.67	3.10	3.50	3.17	1.97	2.23	2.00
40	3.33	3.73	3.33	3.23	3.70	3.33	2.23	2.60	2.33
41	3.33	3.73	3.33	3.47	3.90	3.50	2.53	2.87	2.63
42	3.33	3.73	3.33	3.57	4.03	3.63	2.53	2.87	2.63
43	3.30	3.77	3.30	3.67	4.13	3.73	2.83	3.17	2.93
44	3.30	3.77	3.30	3.73	4.23	3.77	2.97	3.33	3.07
45	3.30	3.77	3.30	3.77	4.20	3.80	3.10	3.53	3.23
46	3.30	3.77	3.30	3.77	4.20	3.80	3.33	3.80	3.43
47	3.30	3.77	3.30	3.77	4.20	3.80	3.50	3.97	3.60
49	3.30	3.77	3.30	3.77	4.20	3.80	3.73	4.20	3.80
50	3.30	3.77	3.30	3.77	4.20	3.80	3.80	4.27	3.90
51	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
52	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
53	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
54	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
55	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
56	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
57	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
58	3.30	3.77	3.30	3.77	4.20	3.80	3.83	4.27	3.93
60	3.30	3.77	3.30	3.77	4.20	3.83	3.83	4.30	3.93
61	3.30	3.77	3.30	3.77	4.20	3.83	3.83	4.30	3.93

Table B-7: The average over three soybeans of leaf width data (N9-N11).

Date	N9 LW	N9 MW	N9 RW	N10 LW	N10 MW	N10 RW	N11 LW	N11 MW	N11 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.03	0.07	0.03	0.00	0.00	0.00	0.00	0.00	0.00
33	0.03	0.07	0.03	0.00	0.00	0.00	0.00	0.00	0.00
34	0.03	0.07	0.03	0.00	0.00	0.00	0.00	0.00	0.00
35	0.13	0.20	0.13	0.00	0.00	0.00	0.00	0.00	0.00

36	0.17	0.23	0.17	0.00	0.00	0.00	0.00	0.00	0.00
37	0.20	0.27	0.20	0.00	0.00	0.00	0.00	0.00	0.00
38	0.30	0.33	0.30	0.00	0.00	0.00	0.00	0.00	0.00
39	0.47	0.53	0.50	0.00	0.00	0.00	0.00	0.00	0.00
40	0.70	0.83	0.77	0.17	0.17	0.17	0.00	0.00	0.00
41	1.20	1.37	1.27	0.20	0.20	0.20	0.00	0.00	0.00
42	1.20	1.37	1.27	0.20	0.20	0.20	0.00	0.00	0.00
43	2.10	2.33	2.20	0.53	0.63	0.57	0.17	0.17	0.17
44	2.40	2.70	2.53	0.90	1.10	1.00	0.30	0.33	0.30
45	2.60	2.93	2.80	1.30	1.53	1.43	0.40	0.37	0.43
46	2.77	3.10	2.97	1.80	2.17	2.03	0.80	0.77	0.77
47	2.80	3.13	3.00	2.17	2.53	2.33	1.20	1.13	1.20
49	3.00	3.37	3.20	2.53	2.97	2.80	2.47	2.53	2.50
50	3.27	3.63	3.50	2.57	3.13	2.90	2.53	2.50	2.47
51	3.43	3.83	3.70	2.57	3.37	3.00	2.73	2.77	2.70
52	3.60	4.00	3.87	2.57	3.43	3.03	2.80	2.83	2.77
53	3.73	4.13	4.00	2.57	3.53	3.17	2.87	2.93	2.83
54	3.80	4.23	4.07	2.57	3.80	3.43	2.97	3.00	2.90
55	3.80	4.23	4.10	2.57	4.00	3.70	3.03	3.07	2.97
56	3.80	4.27	4.10	2.57	4.27	3.90	3.13	3.17	3.07
57	3.80	4.27	4.10	2.57	4.27	3.90	3.13	3.17	3.07
58	3.80	4.23	4.10	2.57	4.57	4.20	3.43	3.50	3.40
60	3.87	4.30	4.17	2.60	4.63	4.27	3.83	3.93	3.73
61	3.87	4.30	4.17	2.60	4.63	4.27	3.83	3.93	3.73

Table B-8: The average over three soybeans of leaf width data (N12-N14).

Date	N12 LW	N12 MW	N12 RW	N13 LW	N13 MW	N13 RW	N14 LW	N14 MW	N14 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.27	0.23	0.23	0.00	0.00	0.00	0.00	0.00	0.00
49	1.27	1.30	1.23	0.33	0.33	0.37	0.10	0.10	0.10
50	1.07	1.13	1.13	0.37	0.37	0.37	0.10	0.10	0.10
51	1.47	1.50	1.57	0.47	0.47	0.50	0.17	0.20	0.17
52	1.80	1.87	1.90	0.67	0.70	0.73	0.20	0.23	0.20
53	2.07	2.17	2.20	1.00	1.03	1.03	0.37	0.30	0.37
54	2.30	2.43	2.40	1.47	1.50	1.47	0.50	0.47	0.50
55	2.37	2.50	2.50	1.80	1.80	1.77	0.73	0.77	0.77
56	2.43	2.60	2.57	1.90	2.13	1.80	1.07	1.10	1.13
57	2.40	2.60	2.57	1.90	2.17	1.83	1.23	1.27	1.27
58	2.53	2.60	2.63	2.43	2.33	2.27	1.73	1.73	1.83
60	2.67	3.00	2.87	2.50	2.43	2.33	2.07	2.13	2.17
61	2.90	3.17	3.00	2.50	2.47	2.33	2.10	2.18	2.20

Table B-9: The average over three soybeans of leaf width data (N15-N17).

Date	N15 LW	N15 MW	N15 RW	N16 LW	N16 MW	N16 RW	N17 LW	N17 MW	N17 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
53	0.10	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00
54	0.17	0.17	0.17	0.00	0.00	0.00	0.00	0.00	0.00
55	0.17	0.17	0.17	0.00	0.00	0.00	0.00	0.00	0.00
56	0.30	0.30	0.27	0.07	0.07	0.07	0.00	0.00	0.00
57	0.30	0.30	0.27	0.07	0.07	0.07	0.00	0.00	0.00
58	0.70	0.63	0.67	0.23	0.23	0.23	0.03	0.03	0.03
60	1.37	1.27	1.30	0.60	0.60	0.63	0.10	0.10	0.10
61	1.50	1.53	1.57	0.85	0.83	0.83	0.13	0.13	0.15

Table B-10: The average over three soybeans of leaf length data (N3-N5).

Date	N3 LW	N3 RW	N4 LW	N4 MW	N4 RW	N5 LW	N5 MW	N5 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	2.53	2.43	0.00	0.00	0.00	0.00	0.00	0.00
9	3.40	3.23	0.00	0.00	0.00	0.00	0.00	0.00
10	3.30	3.87	0.00	0.00	0.00	0.00	0.00	0.00
11	4.27	4.23	0.00	0.00	0.00	0.00	0.00	0.00
12	4.27	4.37	0.00	0.00	0.00	0.00	0.00	0.00
13	4.27	4.43	0.83	1.07	0.83	0.00	0.00	0.00
14	4.33	4.40	1.03	1.30	1.23	0.00	0.00	0.00
15	4.47	4.43	1.37	1.93	1.57	0.00	0.00	0.00
16	4.50	4.50	1.77	2.53	1.97	0.00	0.00	0.00
17	4.53	4.53	2.33	3.13	2.47	0.00	0.00	0.00
18	4.53	4.53	2.73	3.63	2.90	0.00	0.00	0.00
19	4.53	4.57	3.10	3.97	3.20	0.00	0.00	0.00
20	4.53	4.57	3.37	4.13	3.37	0.00	0.00	0.00
21	4.53	4.57	3.48	4.20	3.53	0.93	0.63	0.90
22	4.53	4.57	3.48	4.20	3.53	1.47	1.03	1.43
23	4.37	4.47	3.53	4.23	3.57	1.90	1.37	1.90
24	4.37	4.47	3.53	4.23	3.60	2.60	1.70	2.53
25	4.37	4.47	3.53	4.23	3.60	3.23	2.20	3.20

26	4.37	4.47	3.53	4.23	3.60	3.73	2.57	3.73
27	4.37	4.47	3.53	4.23	3.60	4.07	2.87	4.10
28	4.37	4.47	3.53	4.23	3.60	4.27	3.07	4.37
29	4.37	4.43	3.53	4.23	3.60	4.47	3.20	4.50
30	4.37	4.37	3.50	4.23	3.60	4.53	3.27	4.60
31	4.30	4.33	3.63	4.23	3.60	4.53	3.27	4.63
32	4.27	4.23	3.50	4.23	3.60	4.53	3.27	4.67
33	4.10	4.23	3.50	4.23	3.60	4.53	3.27	4.67
34	3.97	4.17	3.50	4.23	3.60	4.53	3.27	4.67
35	1.33	4.40	3.50	4.23	3.60	4.55	3.27	4.67
36	1.33	4.40	3.50	4.23	3.60	4.53	3.27	4.67
37	1.33	4.40	3.50	4.23	3.60	4.53	3.27	4.67
38	1.33	4.40	3.50	4.23	3.60	4.53	3.27	4.67
39	1.33	4.40	3.50	4.23	3.60	4.53	3.27	4.67
40	1.33	4.40	3.50	4.23	3.60	4.53	3.27	4.67
41	1.33	4.40	3.50	4.23	3.60	4.53	3.27	4.20
42	0.00	2.80	3.50	4.23	3.60	4.53	3.27	4.67
43	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
44	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
45	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
46	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
47	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
49	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
50	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
51	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
52	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
53	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
54	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
55	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
56	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
57	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
58	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53
60	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53

61	0.00	0.00	3.50	4.20	3.57	4.47	3.23	4.53

Table B-11: The average over three soybeans of leaf length data (N6-N8).

Date	N6 LW	N6 MW	N6 RW	N7 LW	N7 MW	N7 RW	N8 LW	N8 MW	N8 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.37	0.53	0.37	0.00	0.00	0.00	0.00	0.00	0.00
25	0.67	0.87	0.67	0.00	0.00	0.00	0.00	0.00	0.00
26	0.80	1.23	0.83	0.00	0.00	0.00	0.00	0.00	0.00
27	1.17	1.63	1.17	0.00	0.00	0.00	0.00	0.00	0.00
28	1.77	2.23	1.63	0.23	0.37	0.23	0.00	0.00	0.00
29	2.20	2.87	2.17	0.37	0.50	0.37	0.00	0.00	0.00
30	2.77	3.47	2.70	0.43	0.57	0.43	0.00	0.00	0.00
31	3.23	4.07	3.17	0.57	0.67	0.57	0.00	0.00	0.00
32	3.77	4.70	3.73	0.73	0.90	0.73	0.20	0.27	0.20
33	4.17	5.13	4.17	1.13	1.47	1.17	0.27	0.40	0.27

34	4.43	5.43	4.40	1.63	2.07	1.60	0.50	0.63	0.50
35	4.60	5.60	4.60	2.40	2.60	2.37	0.77	1.07	0.77
36	4.77	5.73	4.73	3.07	3.70	3.03	0.80	1.50	1.13
37	4.87	5.87	4.83	3.73	4.60	3.73	1.90	2.47	1.87
38	4.87	5.90	4.83	4.13	5.07	4.13	2.43	3.17	2.43
39	4.90	5.93	4.87	4.57	5.70	4.67	3.20	4.07	3.23
40	4.90	5.90	4.87	4.83	6.07	4.90	3.70	4.63	3.67
41	4.90	5.90	4.87	5.03	6.40	5.17	4.13	5.20	4.13
42	4.90	5.90	4.87	5.23	6.67	5.37	4.13	5.20	4.13
43	4.87	5.83	4.83	5.43	6.70	5.50	4.63	5.60	4.67
44	4.87	5.83	4.80	5.53	7.00	5.60	4.87	6.10	4.87
45	4.87	5.83	4.83	5.53	7.03	5.60	5.07	6.40	5.10
46	4.87	5.83	4.83	5.57	7.03	5.63	5.47	6.90	5.47
47	4.87	5.83	4.83	5.57	7.03	5.63	5.73	7.20	5.77
49	4.87	5.83	4.83	5.53	7.00	5.60	6.13	7.63	6.10
50	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.20
51	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
52	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
53	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
54	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
55	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
56	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
57	4.87	5.83	4.83	5.53	7.00	5.60	6.23	7.77	6.23
58	4.87	5.83	4.83	5.57	7.03	5.63	6.23	7.77	6.23
60	4.87	5.83	4.83	5.57	7.03	5.63	6.23	7.77	6.20
61	4.87	5.83	4.83	5.57	7.03	5.63	6.23	7.77	6.20

Table B-12: The average over three soybeans of leaf length data (N9-N11).

Date	N9 LW	N9 MW	N9 RW	N10 LW	N10 MW	N10 RW	N11 LW	N11 MW	N11 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.07	0.10	0.07	0.00	0.00	0.00	0.00	0.00	0.00
33	0.07	0.10	0.07	0.00	0.00	0.00	0.00	0.00	0.00
34	0.10	0.13	0.10	0.00	0.00	0.00	0.00	0.00	0.00
35	0.27	0.37	0.27	0.00	0.00	0.00	0.00	0.00	0.00
36	0.33	0.47	0.33	0.00	0.00	0.00	0.00	0.00	0.00
37	0.47	0.63	0.50	0.00	0.00	0.00	0.00	0.00	0.00
38	0.67	0.80	0.63	0.00	0.00	0.00	0.00	0.00	0.00
39	1.03	1.33	1.07	0.00	0.00	0.00	0.00	0.00	0.00
40	1.43	1.77	1.43	0.40	0.57	0.40	0.00	0.00	0.00

41	2.27	2.53	2.30	0.53	0.70	0.53	0.00	0.00	0.00
42	2.27	2.53	2.30	0.53	0.70	0.53	0.00	0.00	0.00
43	3.83	4.77	3.90	1.30	1.67	1.20	0.50	0.57	0.50
44	4.40	5.40	4.57	2.07	2.53	2.00	0.77	0.87	0.73
45	4.77	6.07	4.97	2.80	3.43	2.73	1.00	1.20	1.00
46	5.00	6.40	5.23	3.83	4.70	3.80	1.73	2.07	1.70
47	5.10	6.50	5.33	4.53	5.57	4.50	2.47	2.93	2.47
49	5.50	7.07	5.73	4.90	6.60	5.37	4.80	6.00	4.90
50	5.97	7.63	6.17	5.40	7.00	5.63	5.03	6.20	4.97
51	6.33	8.07	6.53	5.37	7.17	5.77	5.47	6.93	5.43
52	6.63	8.43	6.80	5.37	7.37	5.90	5.60	7.00	5.57
53	6.87	8.70	7.07	5.37	7.67	6.17	5.77	7.27	5.77
54	7.07	8.93	7.23	5.37	8.30	6.63	5.87	7.50	5.87
55	7.07	9.00	7.27	5.37	8.90	7.13	6.03	7.63	5.97
56	7.07	9.00	7.23	5.37	9.47	7.57	6.23	7.90	6.13
57	7.07	9.00	7.23	5.37	9.47	7.57	6.23	7.90	6.13
58	7.07	9.00	7.27	5.37	10.23	8.07	6.87	8.80	6.73
60	7.10	9.03	7.30	5.40	10.40	8.17	7.63	9.93	7.43
61	7.10	9.03	7.30	5.40	10.40	8.17	7.63	9.93	7.43

Table B-13: The average over three soybeans of leaf length data (N12-N14).

Date	N12 LW	N12 MW	N12 RW	N13 LW	N13 MW	N13 RW	N14 LW	N14 MW	N14 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.37	0.50	0.37	0.00	0.00	0.00	0.00	0.00	0.00
46	0.57	0.73	0.57	0.00	0.00	0.00	0.00	0.00	0.00
47	0.73	0.87	0.70	0.00	0.00	0.00	0.00	0.00	0.00

49	2.93	3.40	2.60	0.77	1.07	0.77	0.30	0.40	0.30
50	2.60	3.30	2.43	0.73	1.03	0.80	0.23	0.37	0.23
51	3.47	4.37	3.23	1.07	1.53	1.20	0.37	0.53	0.37
52	4.17	5.23	3.90	1.50	2.13	1.67	0.43	0.63	0.43
53	4.80	6.10	4.57	2.13	3.10	2.70	0.67	0.93	0.67
54	5.30	6.80	5.03	2.97	4.43	3.27	1.03	1.50	1.10
55	5.47	7.10	5.20	3.70	5.43	3.93	1.60	2.33	1.70
56	5.60	7.23	5.30	4.20	6.23	4.43	2.23	3.30	2.43
57	5.60	7.23	5.30	4.30	6.33	4.47	2.60	3.80	2.77
58	5.67	7.43	5.37	4.60	6.87	4.80	3.50	5.10	3.70
60	6.07	8.10	5.90	4.73	7.23	4.97	4.23	6.23	4.47
61	6.33	8.55	6.23	4.80	7.37	5.07	4.30	6.37	4.50

Table B-14: The average over three soybeans of leaf length data (N15-N17).

Date	N15 LW	N15 MW	N15 RW	N16 LW	N16 MW	N16 RW	N17 LW	N17 MW	N17 RW
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00
52	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00
53	0.30	0.40	0.30	0.00	0.00	0.13	0.00	0.00	0.00
54	0.33	0.53	0.33	0.00	0.00	0.13	0.00	0.00	0.00
55	0.40	0.67	0.47	0.00	0.00	0.13	0.00	0.00	0.00
56	0.73	1.03	0.67	0.13	0.23	0.13	0.00	0.00	0.00

57	0.73	1.03	0.67	0.13	0.23	0.13	0.00	0.00	0.00
58	1.67	2.27	1.50	0.53	0.77	0.47	0.10	0.20	0.10
60	3.00	4.27	2.80	1.10	1.97	1.30	0.23	0.40	0.23
61	3.50	5.10	3.33	1.57	2.87	1.80	0.37	0.43	0.33

Appendix C

This appendix presents how to use the *PlantVR* software. The *PlantVR* software is called the following icon.

Figure C-1: The main program icon.

The main program is shown as the following window. There are eight pages. First, the L-system page is used to create the physiology of plant. Second, the growth function page is used to set the growth function of each component, such as internode *I*, petiole *P*, apex *A*, leaf *L*, and flower *F*. Third, the component page is used to define the leaf and flower shape of plant. Fourth, the parameter page is used to adjust many

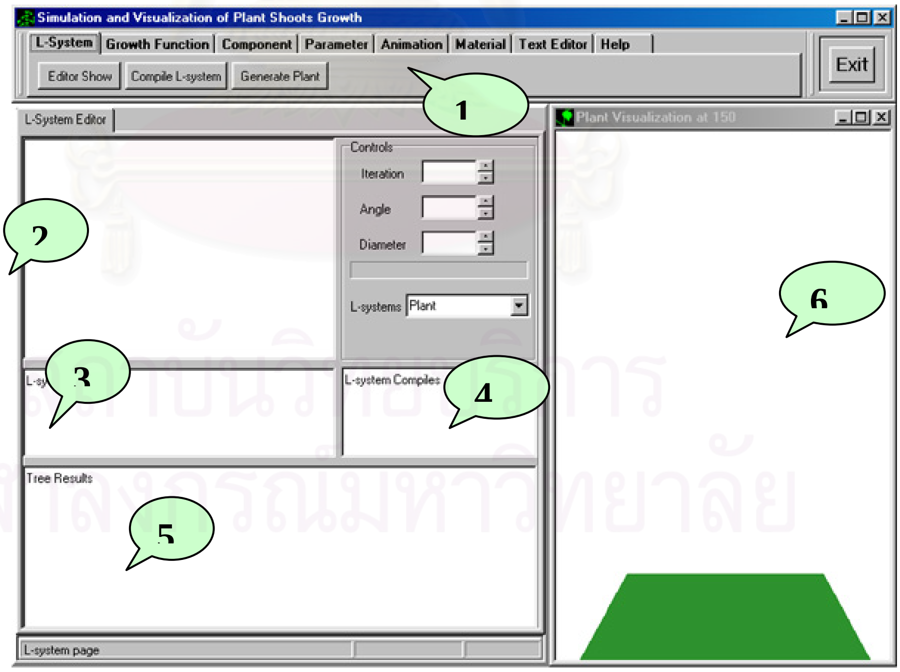


Figure C-2: The main program.

parameters of plant shape and growing. Fifth, the animation page is used to set the animated appearance and export the animation format to “GIF animation”, and the frame of development. Sixth, the material page is used to set the color of each component. Seventh, the text editor is used to store the L-system code. Finally, the help page is user manual.

The description of Figure C-2 has six parts. The first, it is the main menu of the software. The second, it is the L-system editor, the L-system code will be entered in this editor. The third, the L-system result will be shown after compiling the code, the plant prototype. The number of iteration, the angle, the diameter, the production rules, and the endprodution rules are interpreted and writing in the third block. The fourth, the compiled result of L-system is shown in this block. If there are any errors of L-system code, the error message will be shown in this block. The process will be stopped. The fifth, the L-system string will be shown in this block after press the button “Generate Plant” at the main menu. The sixth, the visualization will be shown as three-dimensional plant in this window.

The L-system code is entered in L-system editor page in Figure C-3. It is compiled for the L-system symbol string. The *Soybean* prototype in Chapter 4 will be used for an example model. The L-system is given below.

```
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
}
```

The *Soybean* prototype has six iteration, 45 degrees for the petiole angle, 1.5 centimeters for diameter of first internode, three production rules, and three endproduction rules.

1. The L-system page

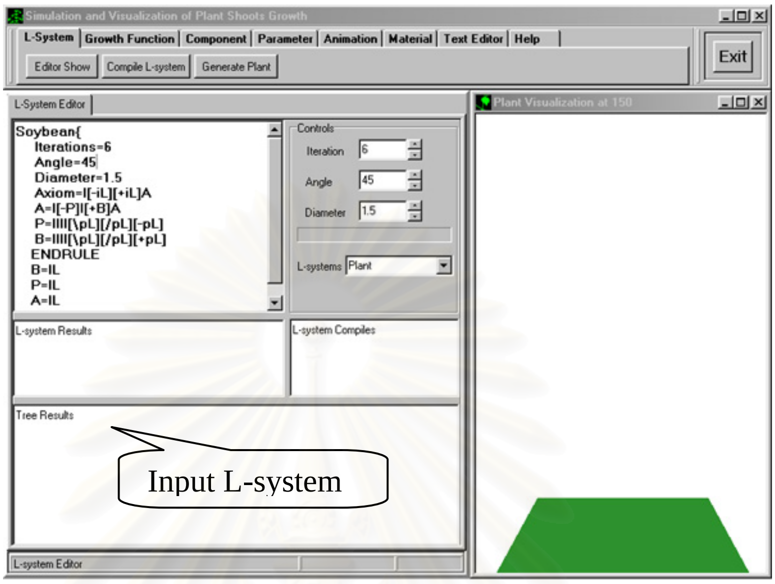


Figure C-3: The L-system input.

The generated plant is constructed by the following step:

1. Input L-system code in the L-system code editor and set the appropriated iteration, angle, and diameter
2. Select the “Compile L-system” button to compile the code and check the syntax
3. Select the “Generate Plant” button to interpret the L-system string to the plant physiology

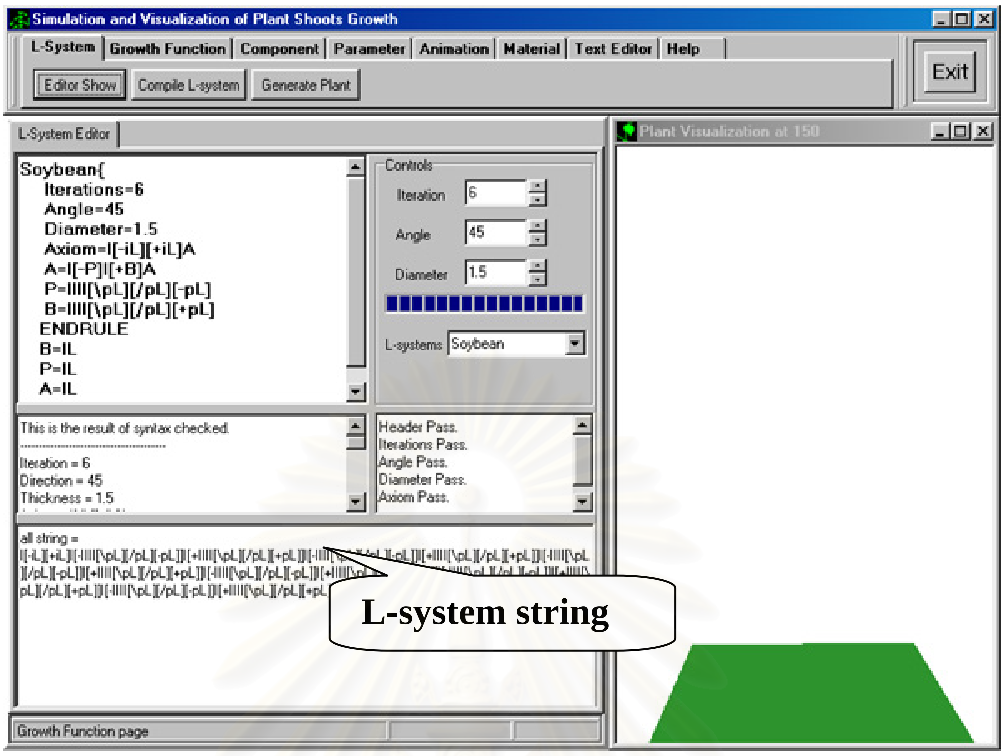


Figure C-4: The result after generating plant.

The final of three steps for creating plant model in L-system page is shown in Figure C-4. If there are any error in each step, the user will edit the L-system code and recompile and regenerate the plant.

2. The growth function page

The growth function page consists of the growth function of the internode, the petiole, the leaf length, the leaf width, the apex, and the flower for the symbol *I*, *i*, *P*, *p*, *L*, *A*, *F*, respectively. The growth function page is shown in Figure C-5. The growth function of each component is selected as follow the symbol of L-system string in L-system page.

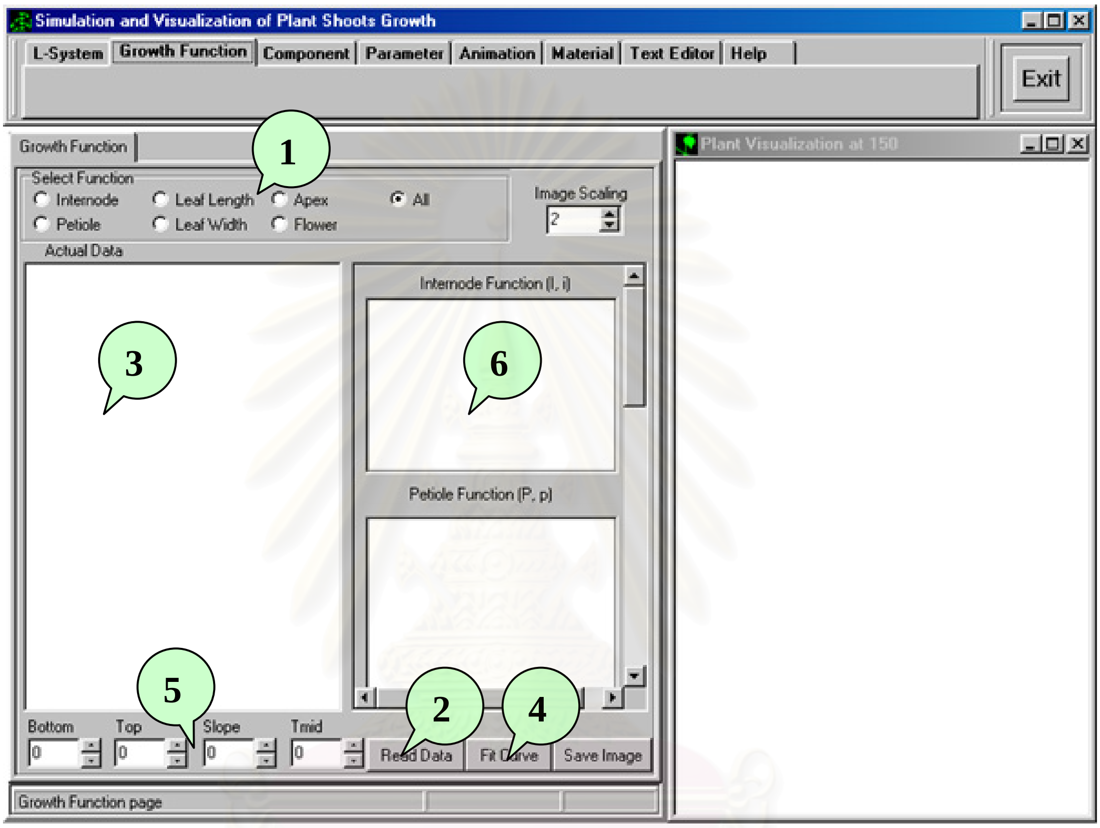


Figure C-5: The growth function page.

The step of growth function is described the following steps:

1. Select the function of symbol such as select the internode radio button for symbol *I* and *i*, the petiole button for symbol *P* and *p*, the leaf length for symbol *L*, the leaf width for symbol *L*, the apex for symbol *A*, the flower for symbol *F*, and all the button for every component.
2. Select the "Read data" button to open the data file, the data will be shown in the callout number three.
3. Select the "Fit Curve" button at callout number four to approximate the growth function of each component.
4. The four parameters of data will be shown at the callout number five. There are bottom, top, slope, and tmid value. Their meanings were described in Chapter 4.

5. The growth function of each component will be drawn in the callout number six

The internode component will be selected and open the data as Figure C-6. The data file of internode are shown in Figure C-7.

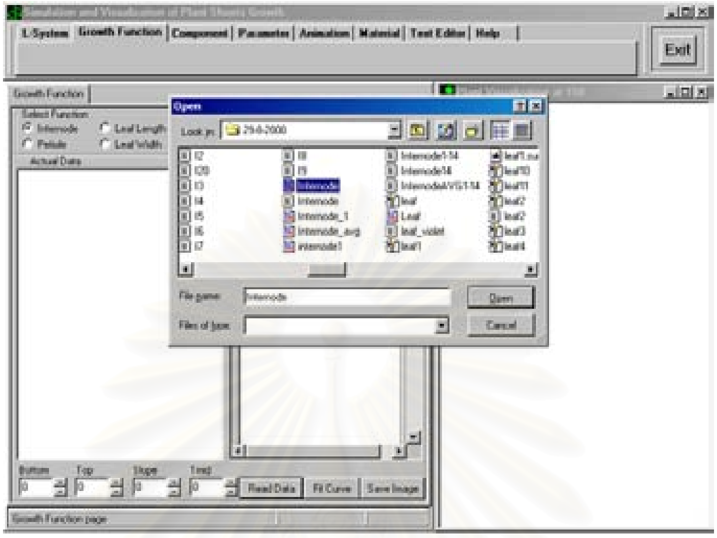


Figure C-6: Open the data file.

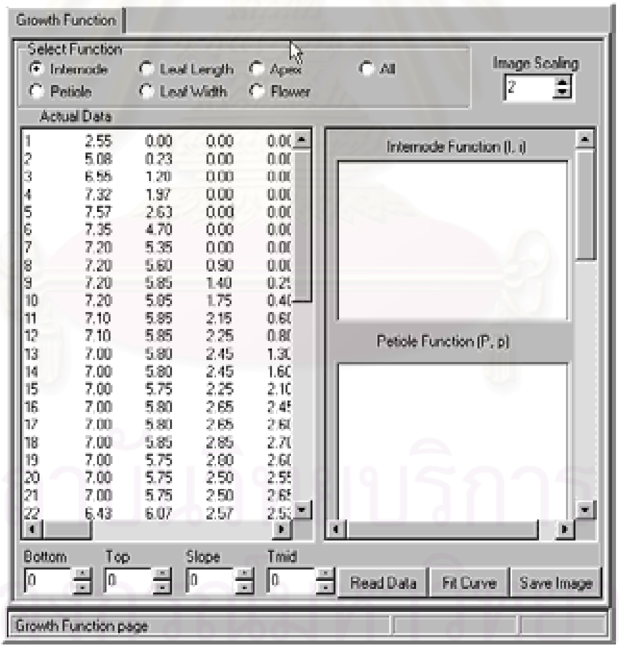


Figure C-7: Internode data.

The curve of internode growth function is shown in Figure C-8. The red line is the average of raw data, the blue line is the approximated growth function. The bottom, the top, the slope, and the tmid are 0.1275, 4.039, 0.65, 7.0, respectively.

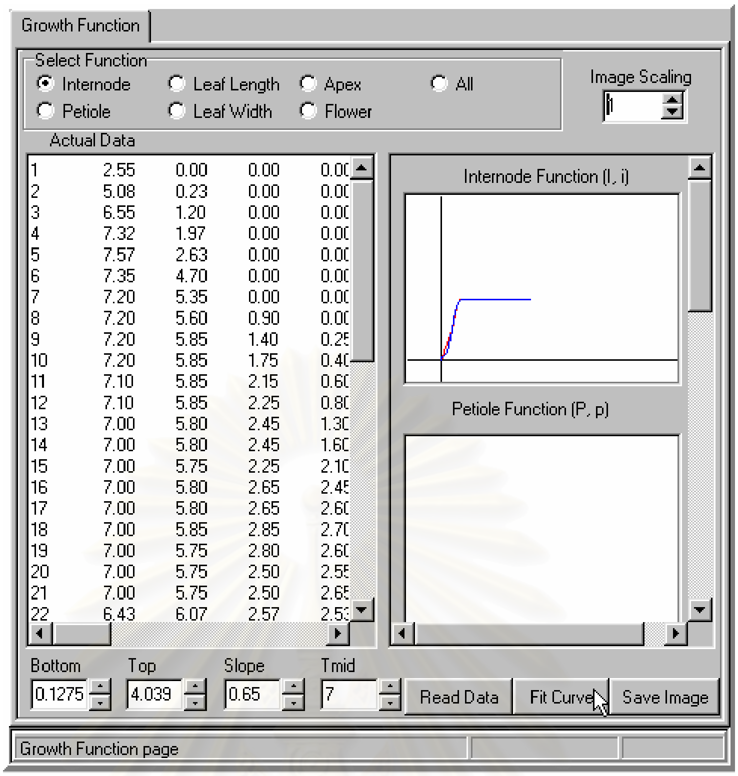


Figure C-8: Internode growth function.

The petiole data will be selected like the internode, that is, select the petiole radio button, and select the "Read data" button. The data file of petiole will be shown like Figure C-9. Then click the "Open" button to open the petiole data file.

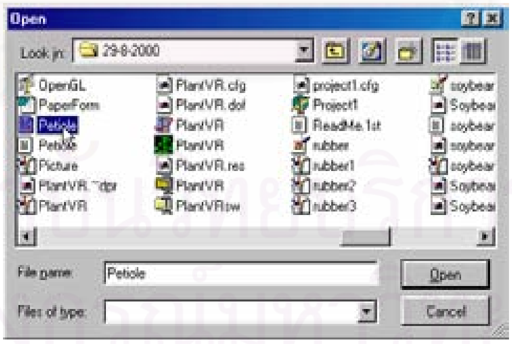


Figure C-9: Open the petiole data file.

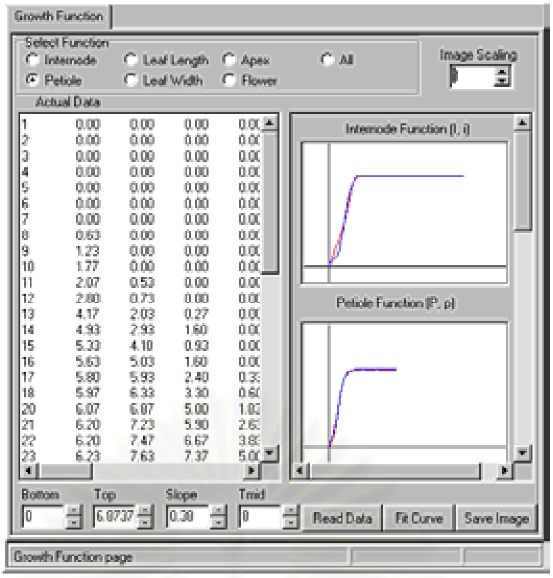


Figure C-10: Petiole growth function.

Then, click the "Fit Curve" button to approximate and draw the curve of petiole growth function in Figure C-10.

To open the leaf length data, click the "Read data" and select the leaf length data file and click "Open" button like Figure C-11, then click the "Fit curve" button to show the curve of growth function as Figure C-12.

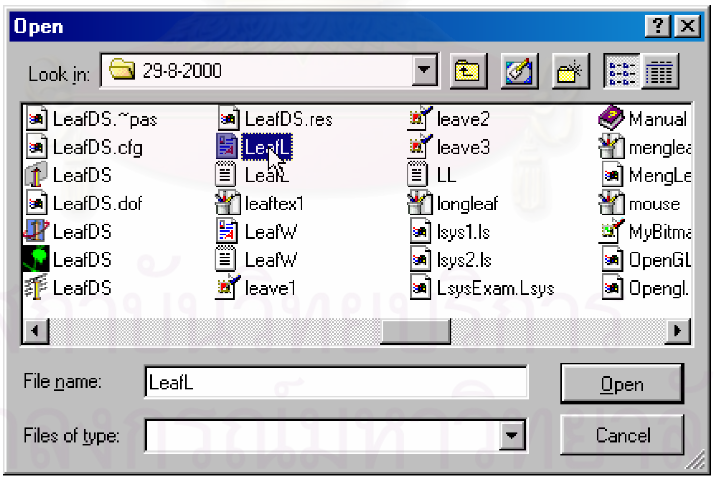


Figure C-11: Open the length of leaf data.

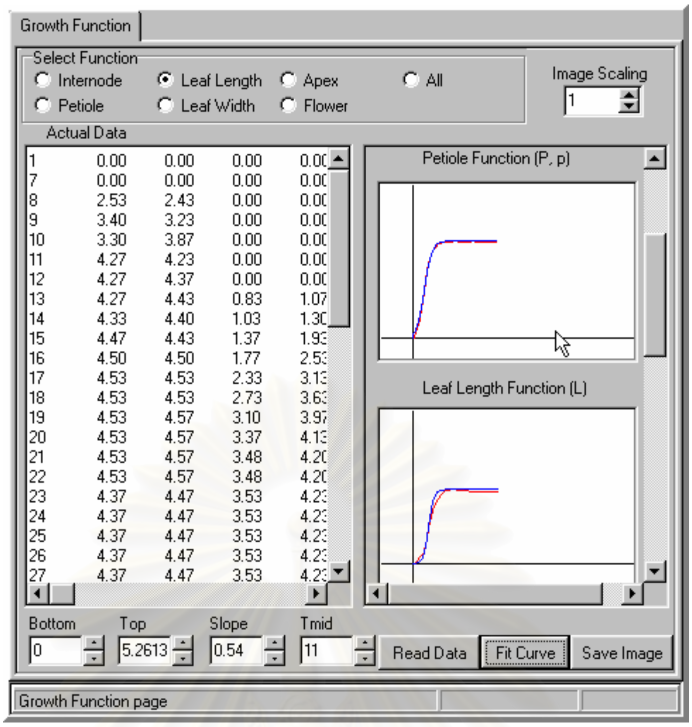


Figure C-12: The leaf length growth function.

In a similar way, click the "Read data" button to open the leaf width data and click "Open" button as Figure C-13, and click the "Fit Curve" button to fit the growth curve of leaf width as Figure C-14.

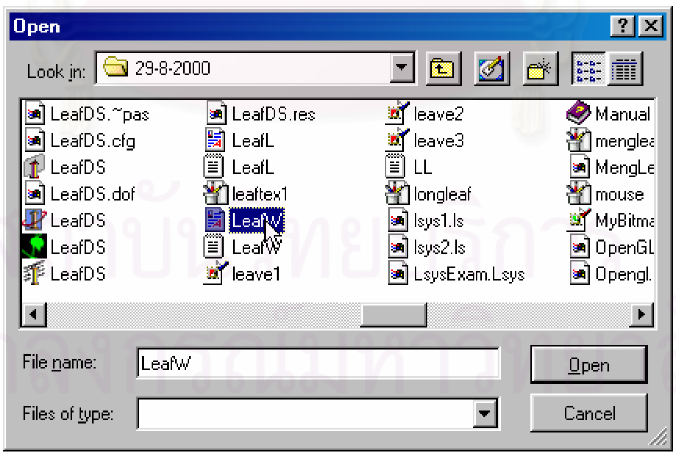


Figure C-13: Open the width of leaf data.

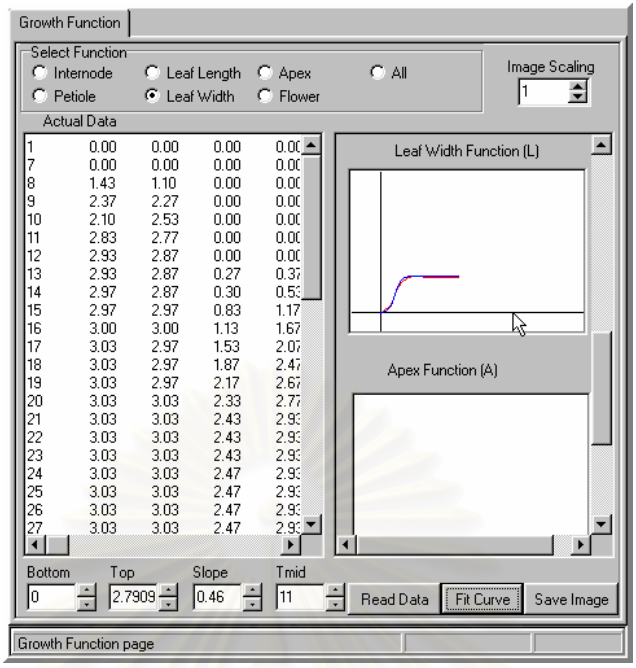


Figure C-14: The leaf width growth function.

The apex represents with the symbol *A*, therefore, we must select and set the growth function for apex growth function. Click the "Read data" button to open the data file, we assume that the apex growth curve as same as the internode function. Then click the internode file and fit curve, the result of apex growth function is shown in Figure C-15.

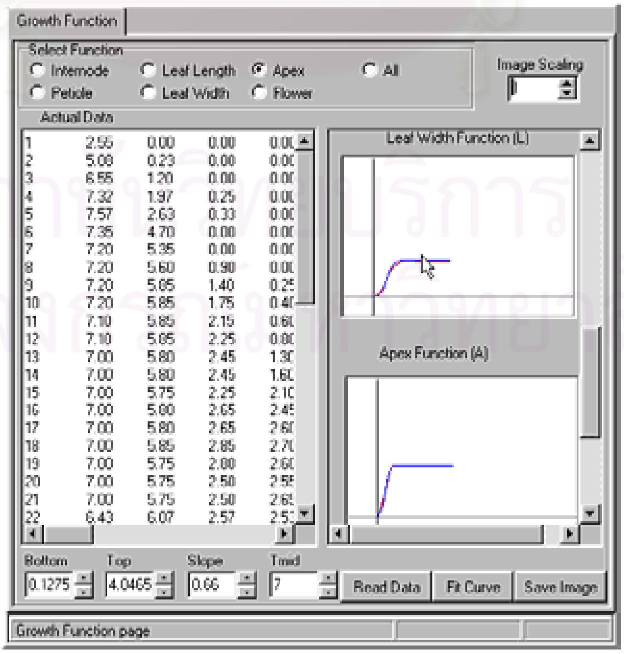


Figure C-15: The apex growth function.

All of growth function , internode, petiole, leaf length, leaf width, and apex are summarized in Figure C-16.

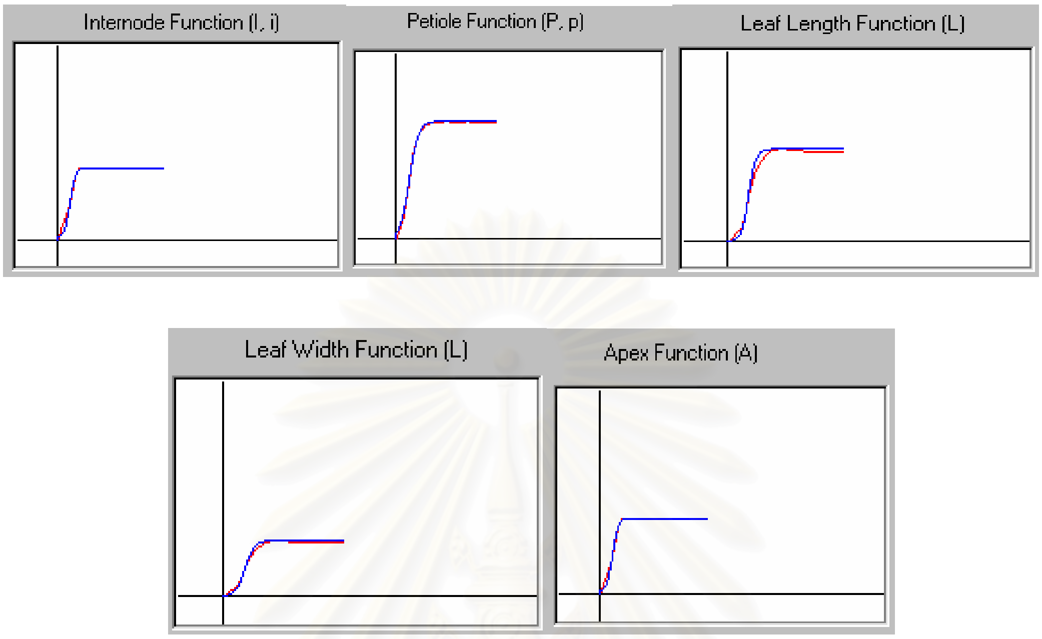


Figure C-16: The growth function of internode, petiole, leaf length , leaf width, and apex function.

3. The component page

The component page is used to define the shape of leaf and flower. To define the leaf and the flower component, the order of definition is shown in Figure C-17. First, Click "Open 3D Surface" to open the leaf and the flower library as Figure C-18. Second, click the radio button "Leaf Surface" to define the leaf surface. Third, select the leaf type at the combox box like Figure C-19 and the leaf shape will be shown in the callout number three like Figure C-20. If the letter *F* is appeared in the L-system string, we should define the flower shape. Fourth, click the radio button "Flower Surface" and select the flower type, the flower shape will display at the callout number five.

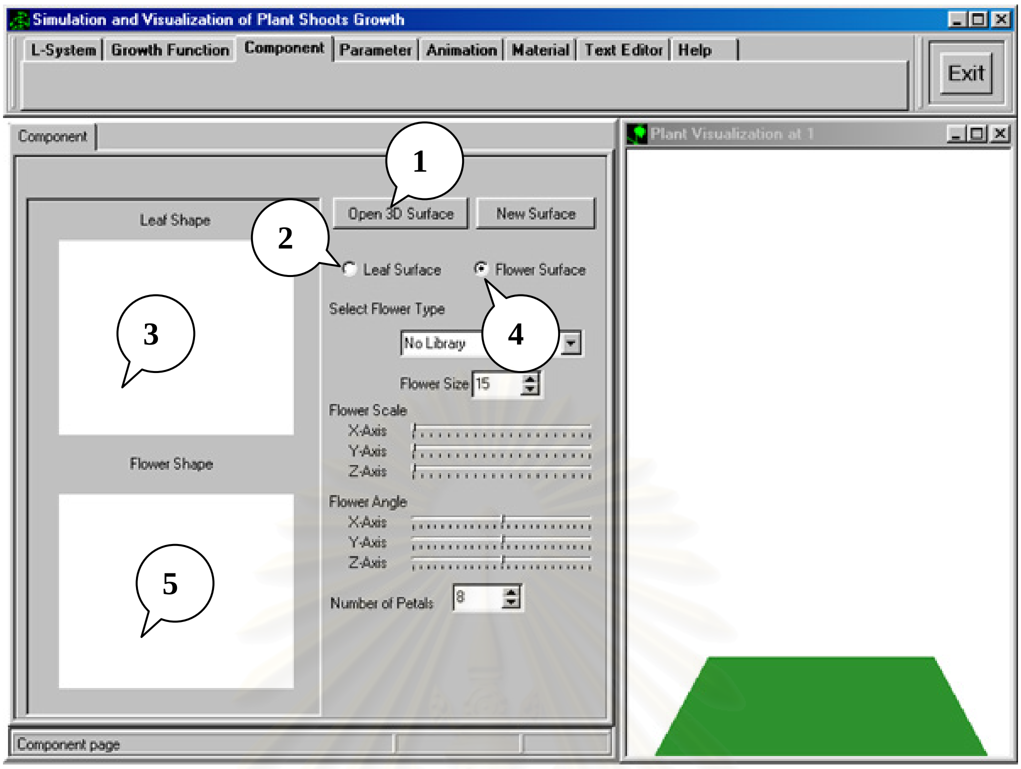


Figure C-17: The component page.

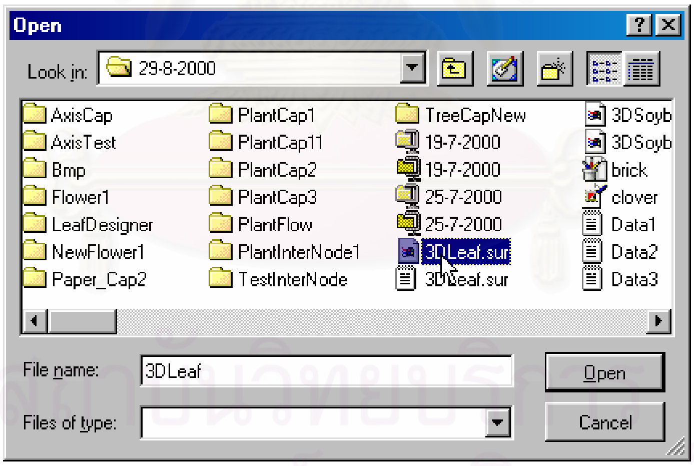


Figure C-18: Open the leaf and flower library.

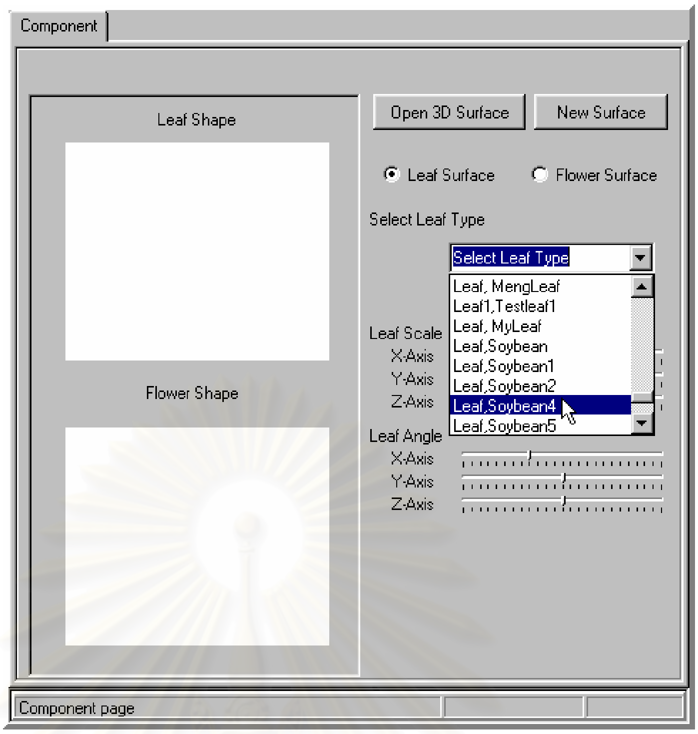


Figure C-19: Select the soybean leaf.

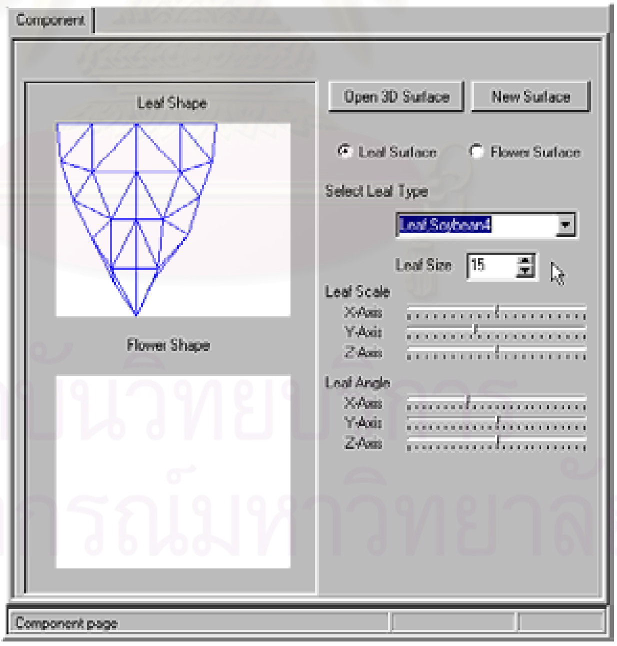


Figure C-20: The part shape of soybean leaf.

4. The parameter page

This page is used to set the plant shape and the plant growth parameter. The parameters are described as following.

1. The initial value of petiole angle is used to set the initial angle of branch and main stem.
2. Stem length is used to control the length of main stem.
3. Node diameter is used to set the diameter of internode and petiole.
4. Node Birth Rate is used to set the initial time of each component.
5. Petiole length is used to control the length of petiole.
6. Petiole bit angle is used to control the angle between the petiole and main stem.
7. Internode reduce is used to control the internode diameter.
8. Petiole reduce is used to control the petiole diameter.
9. Leaf reduce is used to control the leaf size.
10. Flower reduce is used to control the flower size.
11. Short internode ratio is used to control the length ratio between the short internode and the internode.

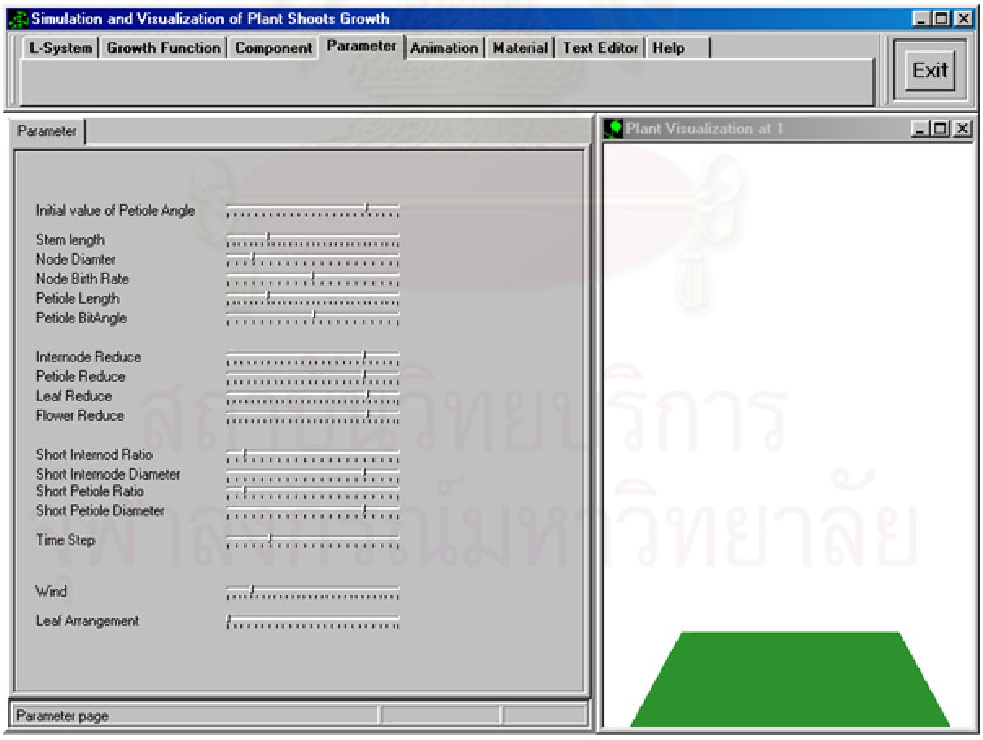


Figure C-21: The parameter page.

12 Short internode diameter is used to control the diameter ratio between the short internode and the internode.

13. Short petiole ratio is used to control the length ratio between the short petiole and the petiole.
14. Short petiole diameter is used to control the diameter ratio between the short petiole

    and the petile.
15. Time stem is used to set the increasing time step.
16. Wind is used to set the bit randomness angle of leaf.
17. Leaf arrangement is used to set the arrangement of leaf angle.

All parameter is shown in the parameter page as Figure C-21.

5. The animation page

The animation page is used to control the animation of plant such as the rotation, translation, zoom, velocity of animation, scaling, capture image. The animation page is shown in Figure C-22.

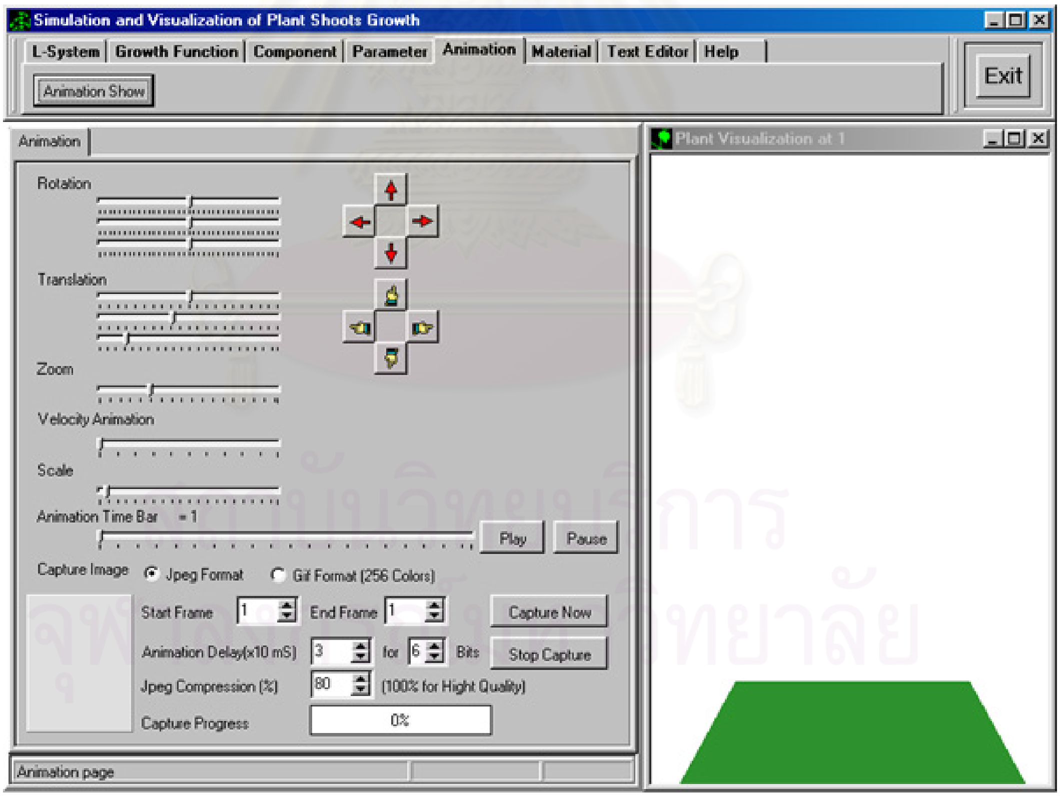


Figure C-22: The animation page.

Figure C-23: The menu of animation page.

To show the plant model, click “Animation Show” button as Figure C-23. The plant model is displayed in the plant Visualization windows as Figure C-24.

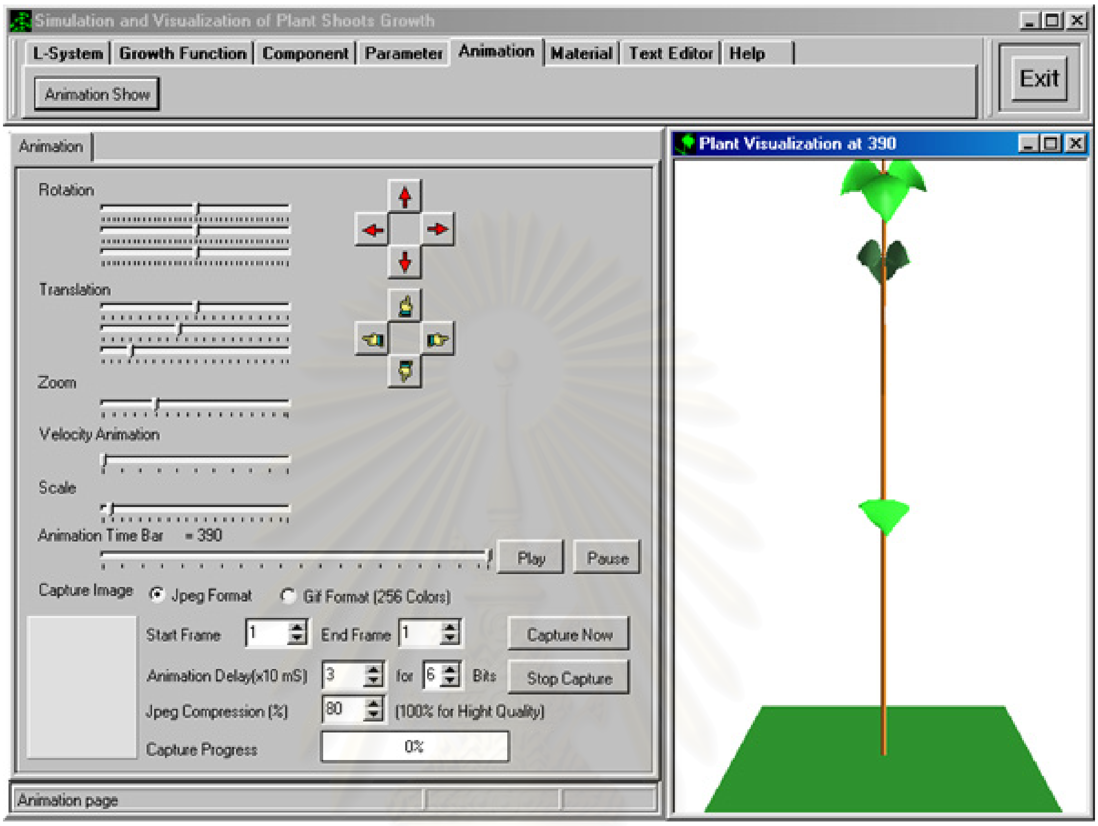


Figure C-24: The first result of plant model.

To reduce the plant main stem, click the stem length to reduce the value at the trackbar as Figure C-25. The plant model is displayed in real-time showing.

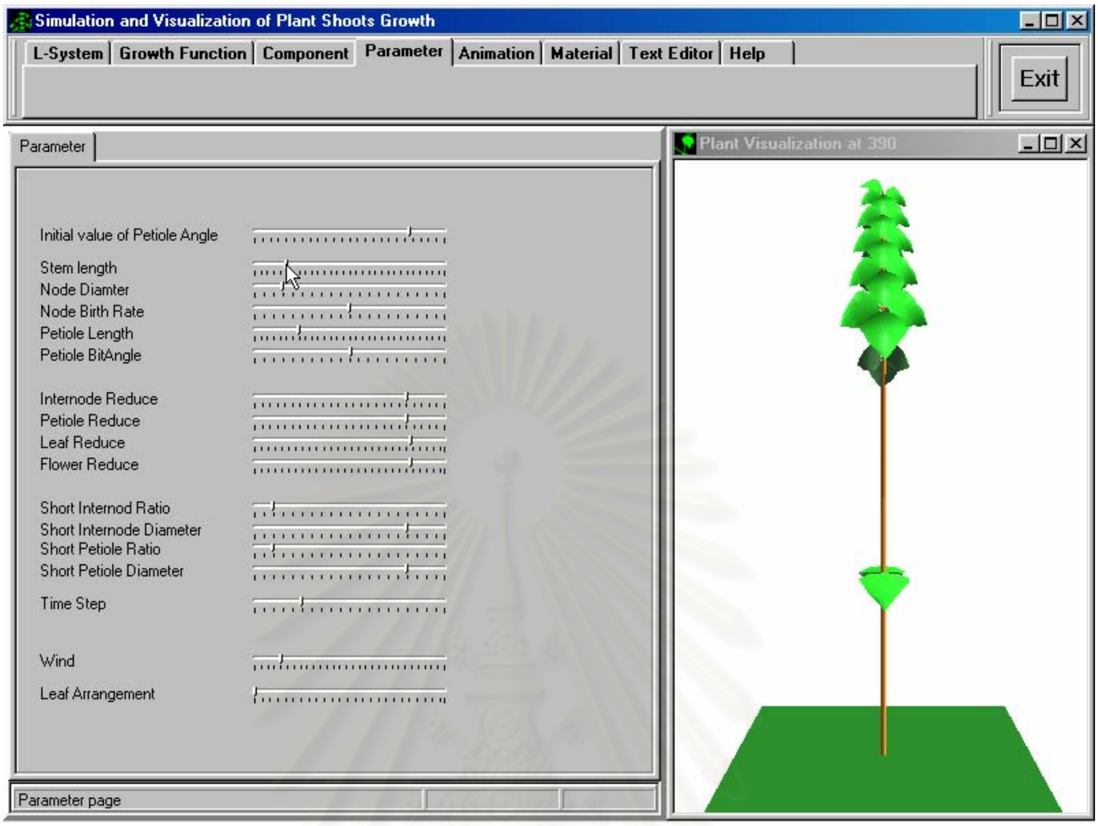


Figure C-25: Set the length of main stem.

To rotate the plant model, click the animation page, and click the second trackbar of rotation to rotate about Y-axis. The plant model is shown in Figure C-26.

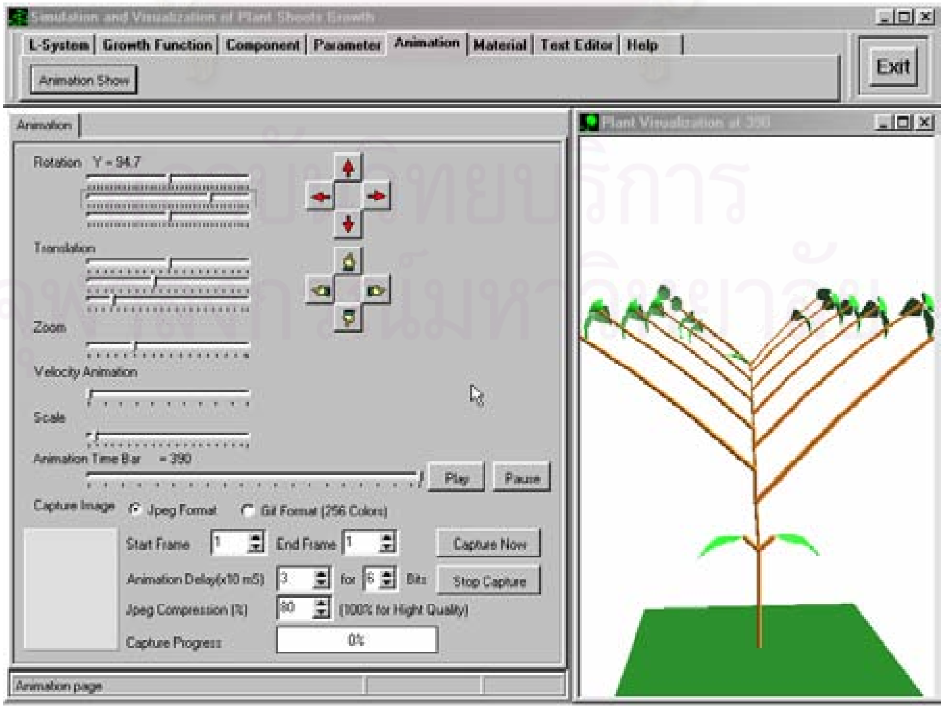


Figure C-26: Adjust the rotation on Y-axis.

To reduce the petiole length, click petiole length. The plant model is displayed as Figure C-27.

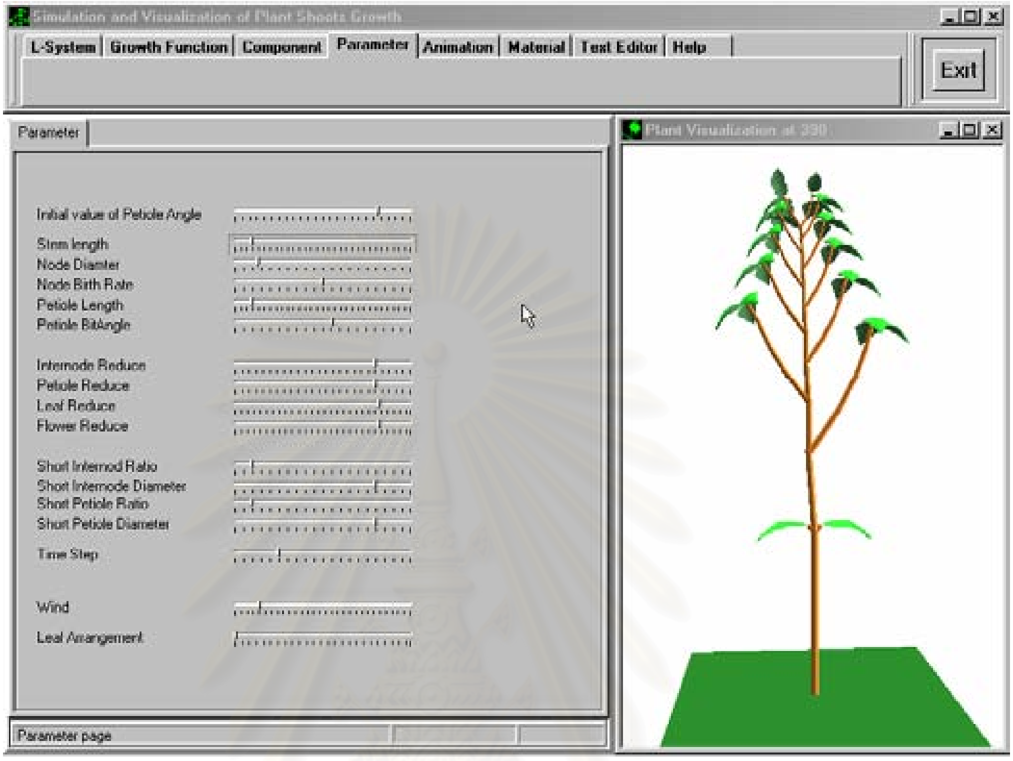


Figure C-27: Adjust the appropriated parameter.

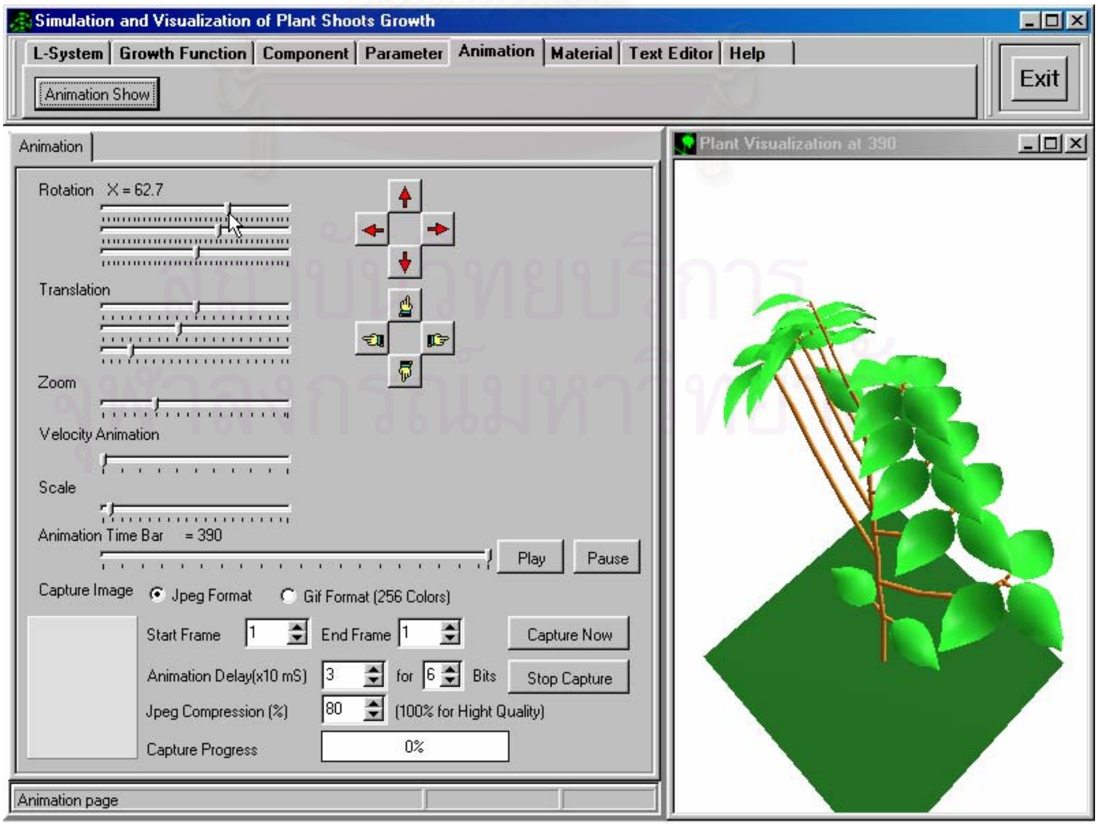


Figure C-28: The top view of plant.

To show the top view of the plant model, click the animation page and click the rotation trackbar. The result will be shown as Figure C-28.

6. The material page

The material page is used to set the color of each component such as the internode, the petiole, front leaf, back leaf, front petal of the flower, back petal of the flower, the background, and the land. Especially, the used can design the leaf texture in order to map the leaf. The material page is shown in Figure C-29.

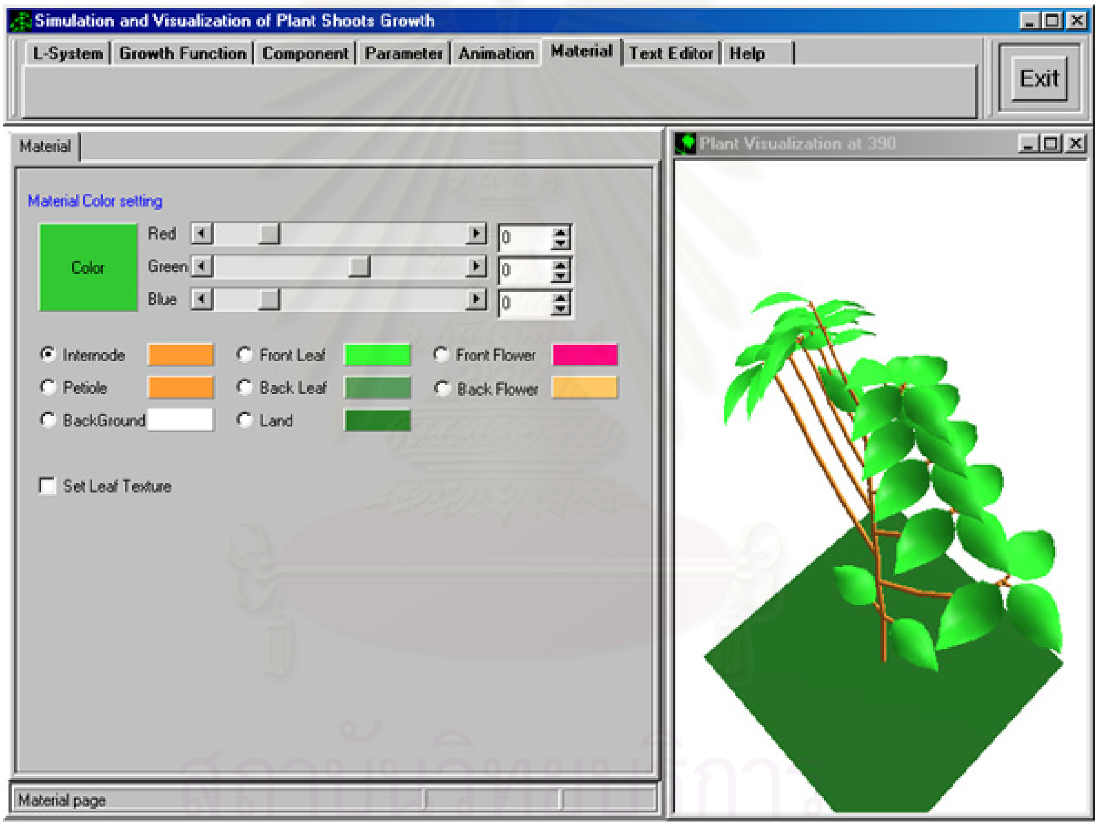


Figure C-29: The material page.

To set the leaf texture, click the check box "Set Leaf Texture" as Figure C-30.The material page will show the texture information. For example, select the leaf texture number two as Figure C-31. The result of texture mapping displays in Figure C-32.

Figure C-30: Select the texture of leaf .

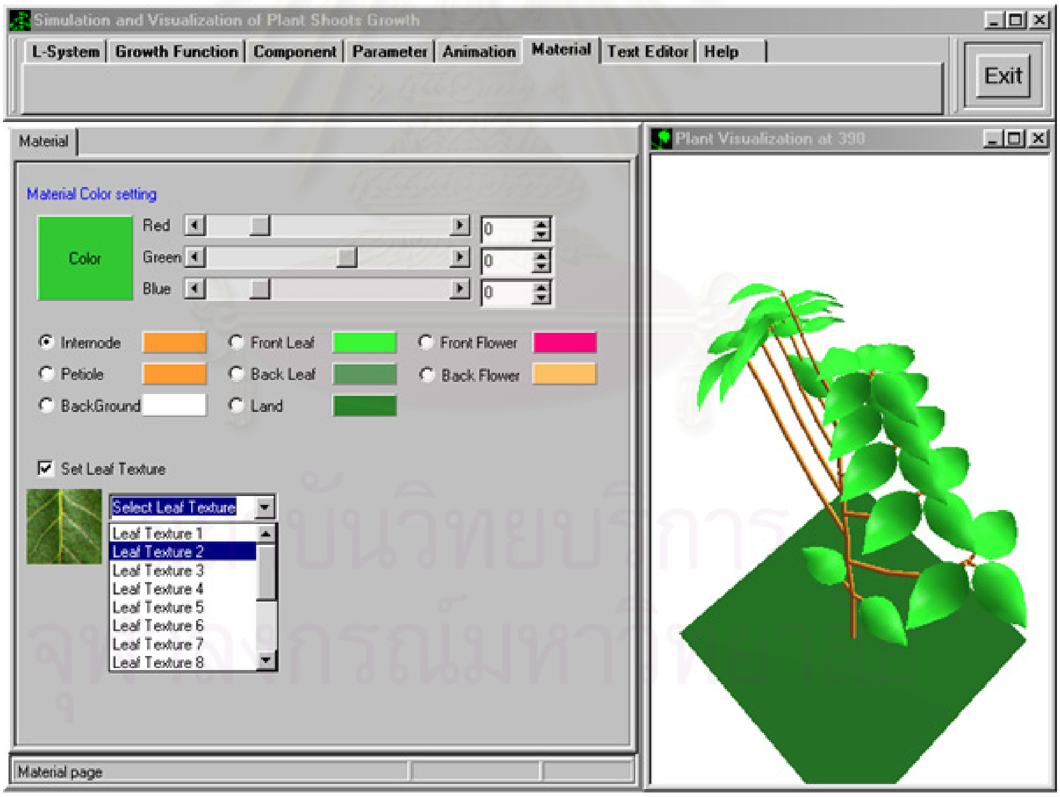


Figure C-31: Select the leaf texture 2.

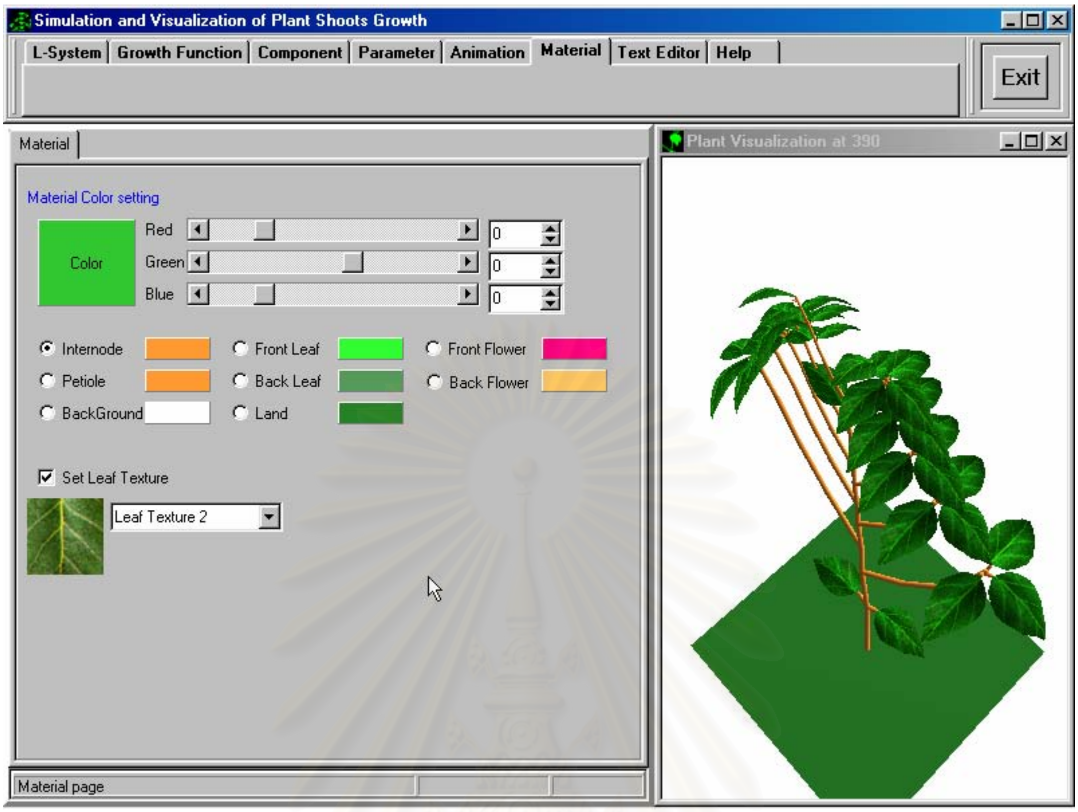


Figure C-32: The result of texture 2.

To open the soybean texture, click the texture image. Select the soybean texture as Figure C-33.

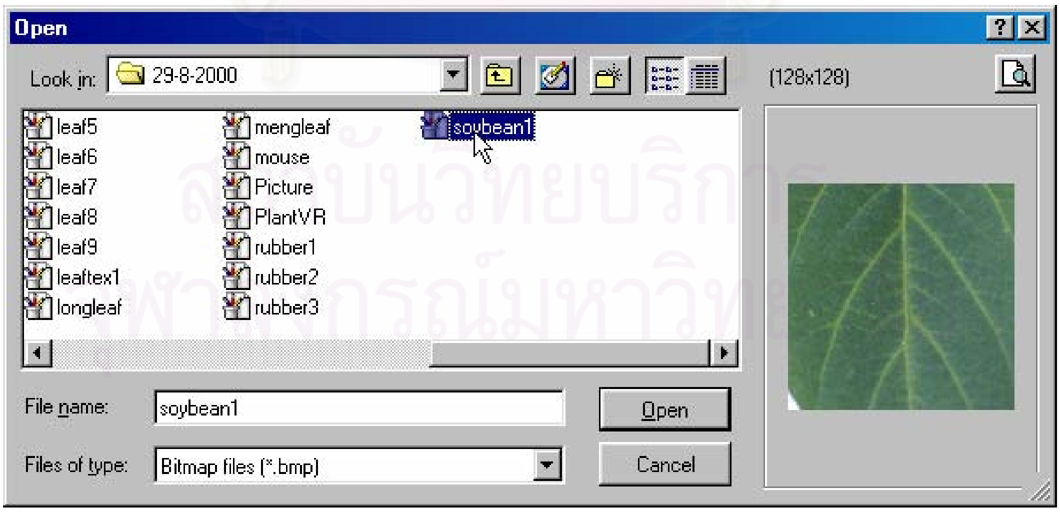


Figure C-33: Open soybean texture.

Click “Open” button. The result displays in the Figure C-34.

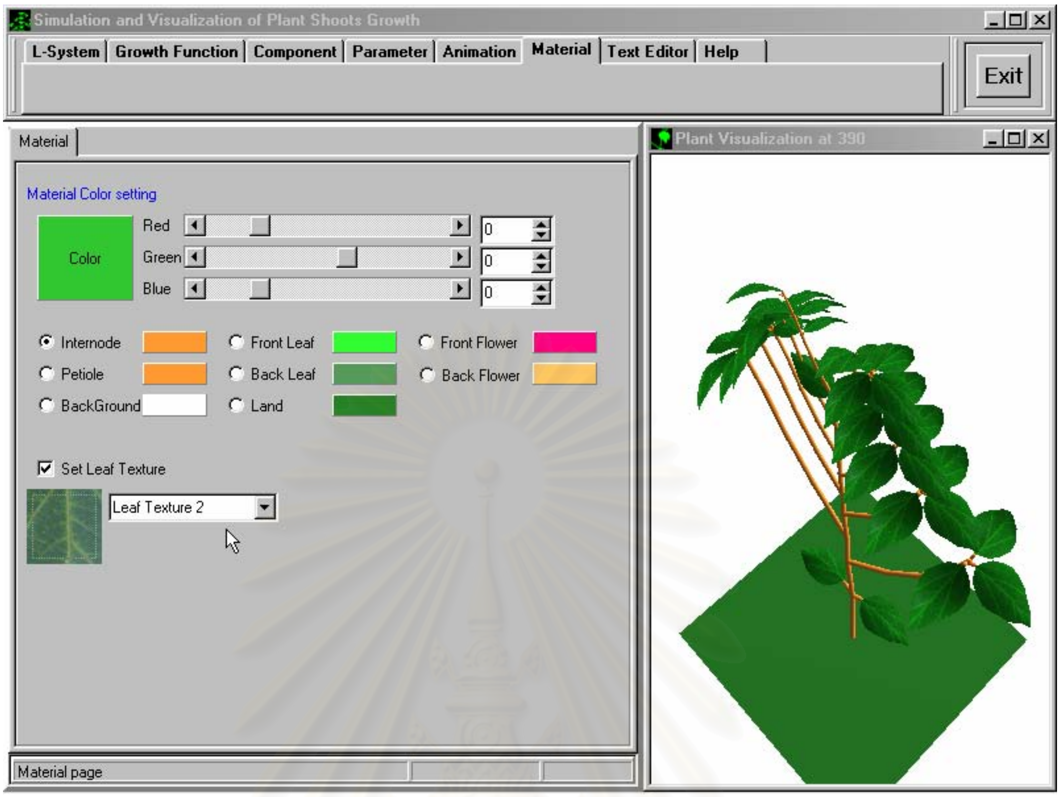


Figure C-34: The result after selecting the soybean texture.

To change the internode color, click the radio button "Internode" and adjust the preferred color. The plant shows the internode color immediately as Figure C-35.

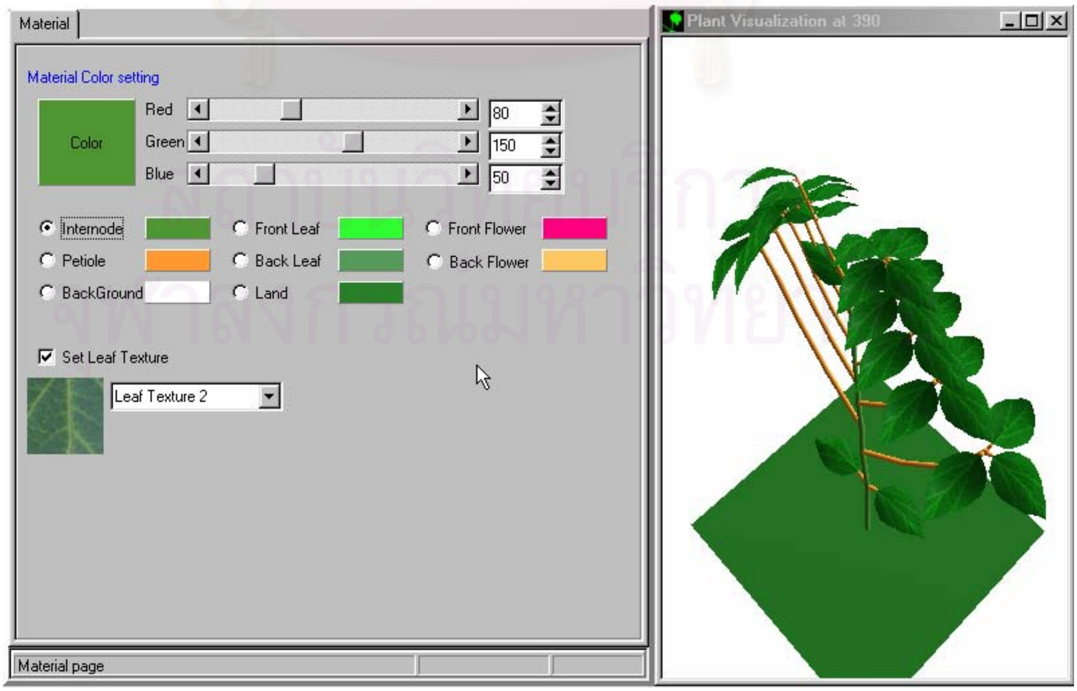


Figure C-35: Set the internode to green color.

Click the radio button "Petiole" and adjust the color of petiole like internode. The petiole color will show in Figure C-36.

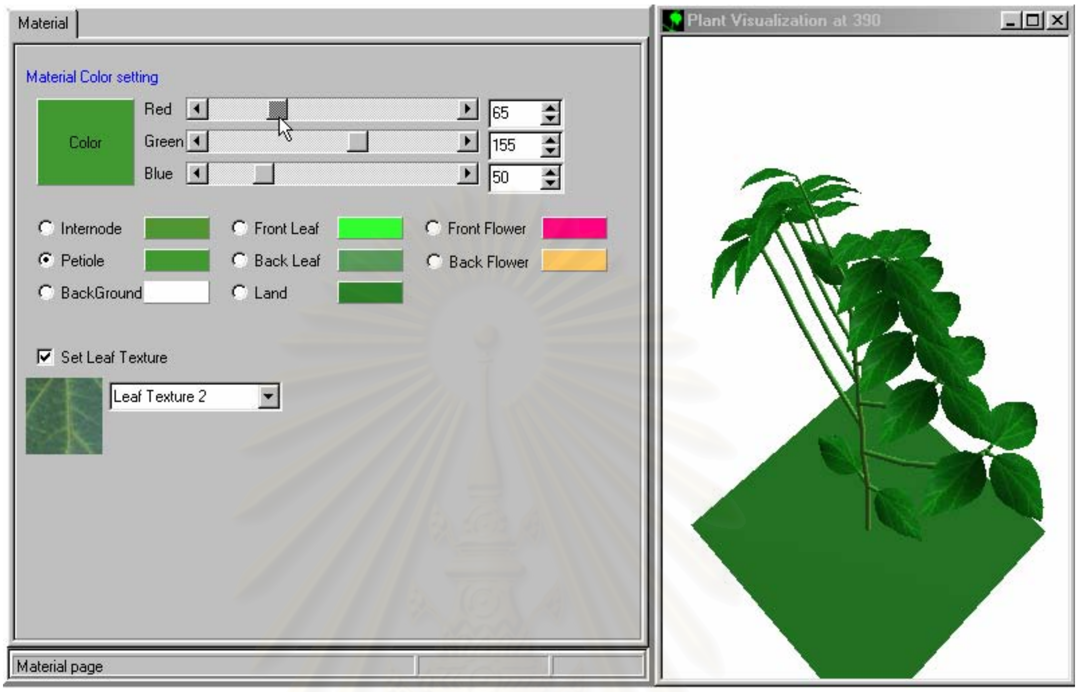


Figure C-36: Set the petiole to green color.

To adjust the perspective view as Figure C-37, click the animation page, and adjust the rotation trackbar.

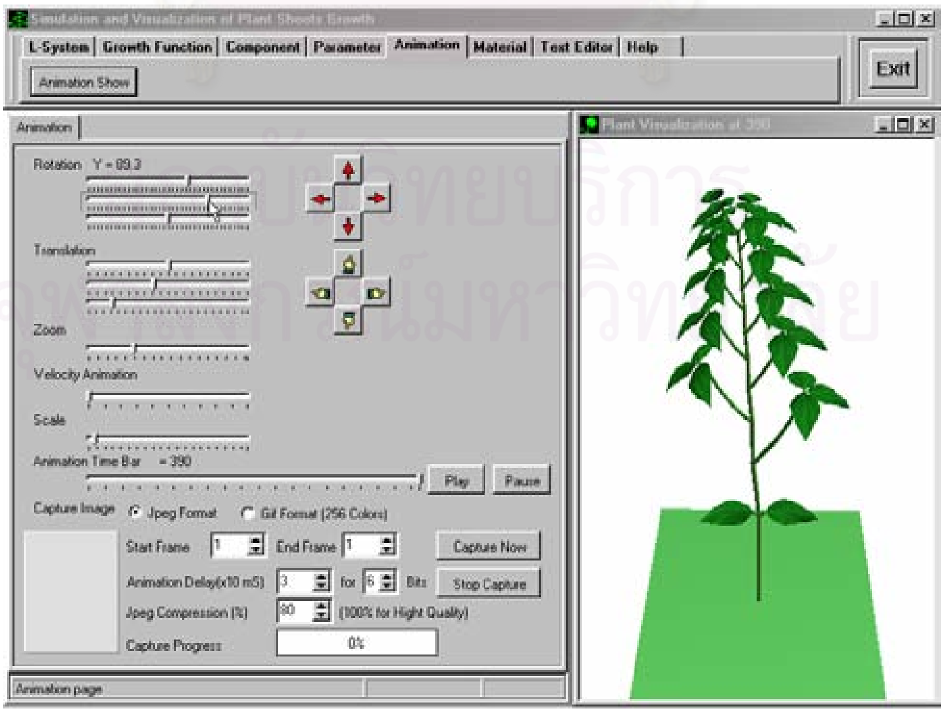


Figure C-37: Adjust the new perspective view.

To show the plant development at any time step, click the trackbar "Animation Time Bar". For example, adjust the animation time bar at 19, the plant development is shown in Figure C-38.

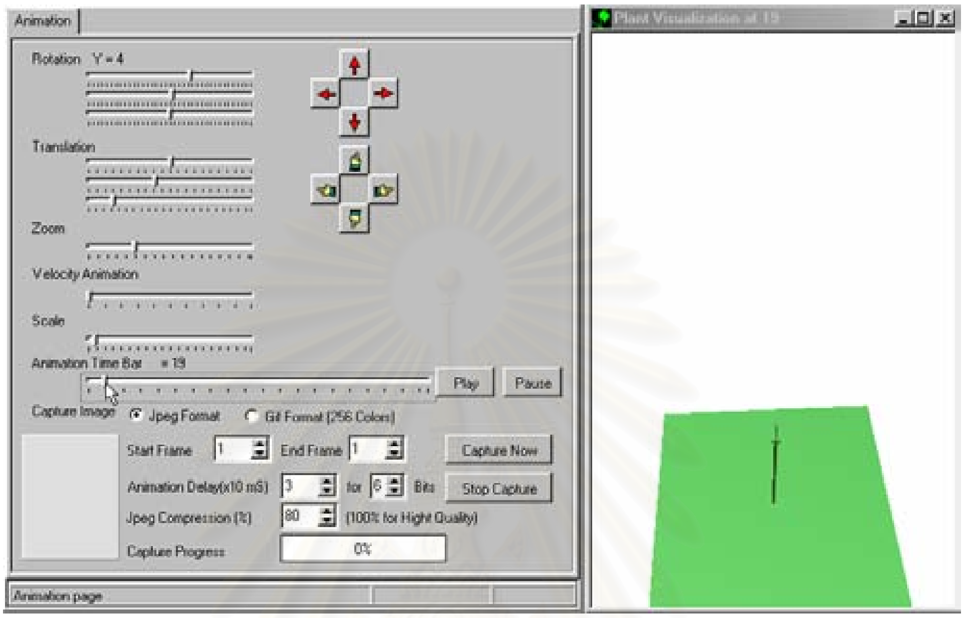


Figure C-38: The animation of plant growth at time 19.

At the time 53, the plant development is displayed as Figure C-39.

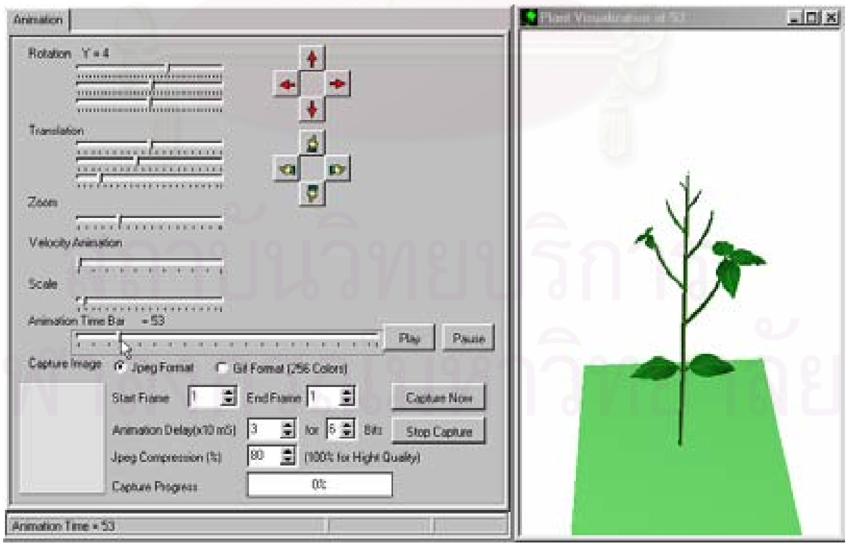


Figure C-39: The animation of plant growth at time 53.

At the time 75, the plant growth is shown in Figure C-40.

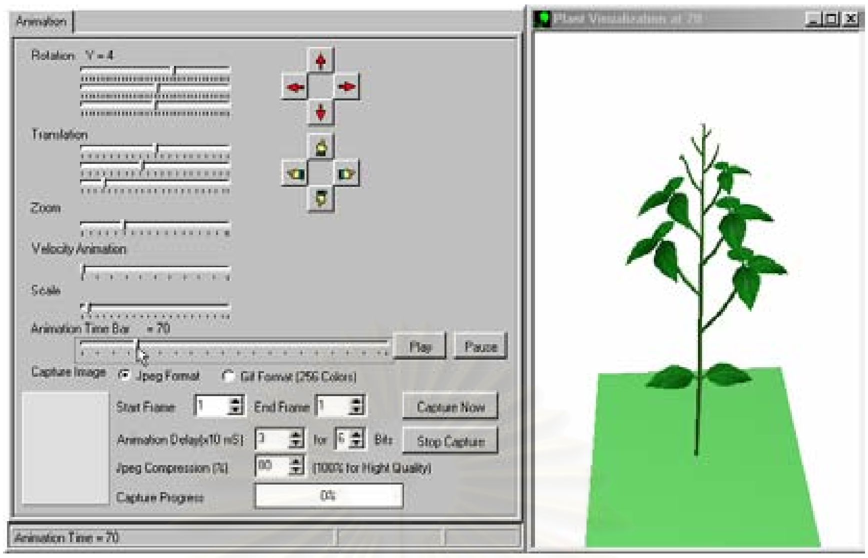


Figure C-40: The animation of plant growth at time 75.

At the time 172, the result will be shown in Figure C-41.

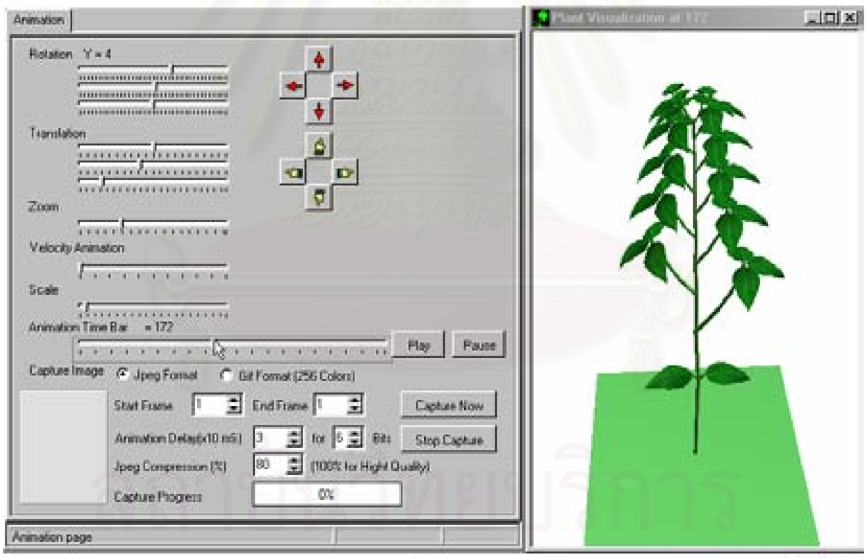


Figure C-41: The animation of plant growth at time 172.

To export the animation output frame and the animation output file like GIF animation, adjust the appropriated parameter such as the animation delay (set the value to one for fastest) and the quality of the animation file (set the value to eight for high quality of GIF animation). Set the capture image type JPEG format or GIF format and the JPEG compression quality (set to 100 for high quality).

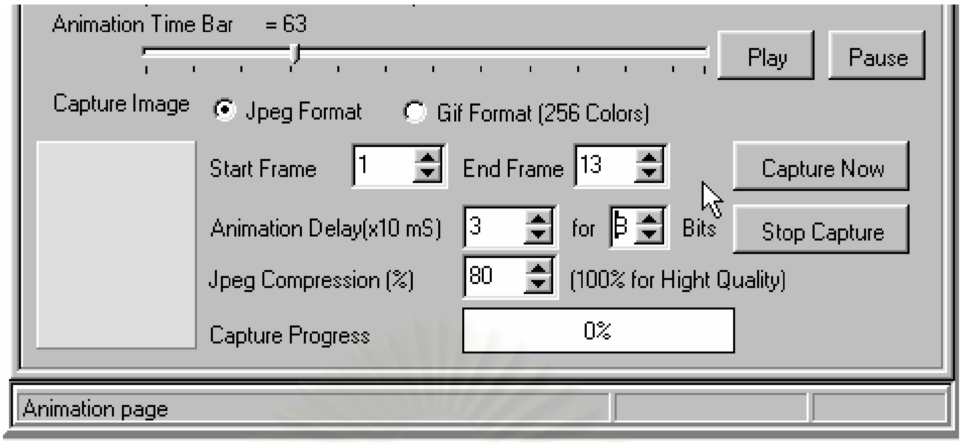


Figure C-42: The capture setting.

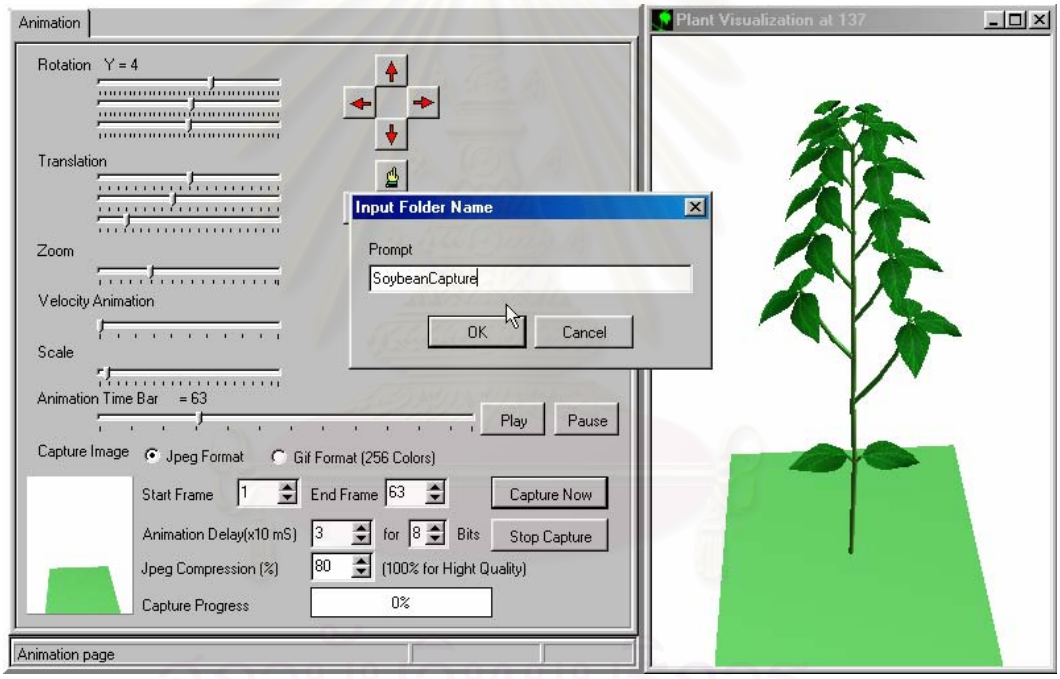


Figure C-43: Input the target folder of animation frames.

The animation is capturing using the plant visualization window. Don't open the other window that makes this window unclear. The user can set any parameter to include the changing to the output file such as rotation the plant model.

A snap short at sixty-nine percent is shown in Figure C-44. If you want to stop the animation, click the "Stop Capture" button.

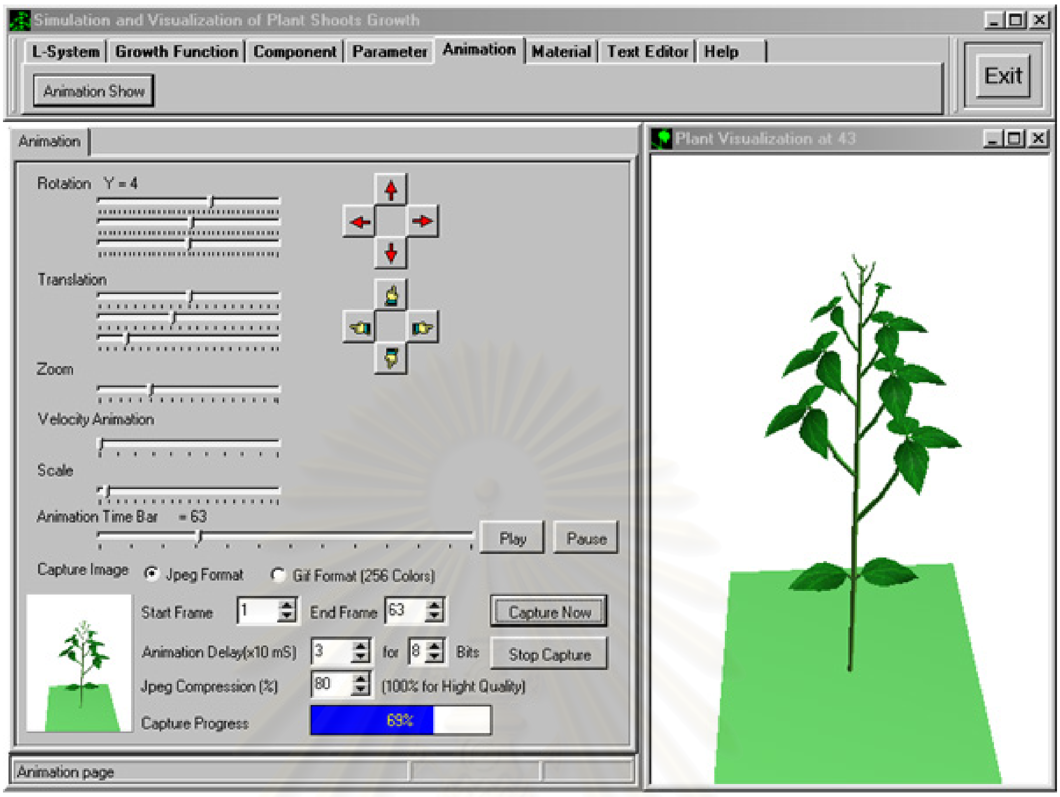


Figure C-44: Capturing at sixty-nine percent.

After the process has finished, the program shows the message "Capture Completed" as FigureC-45.

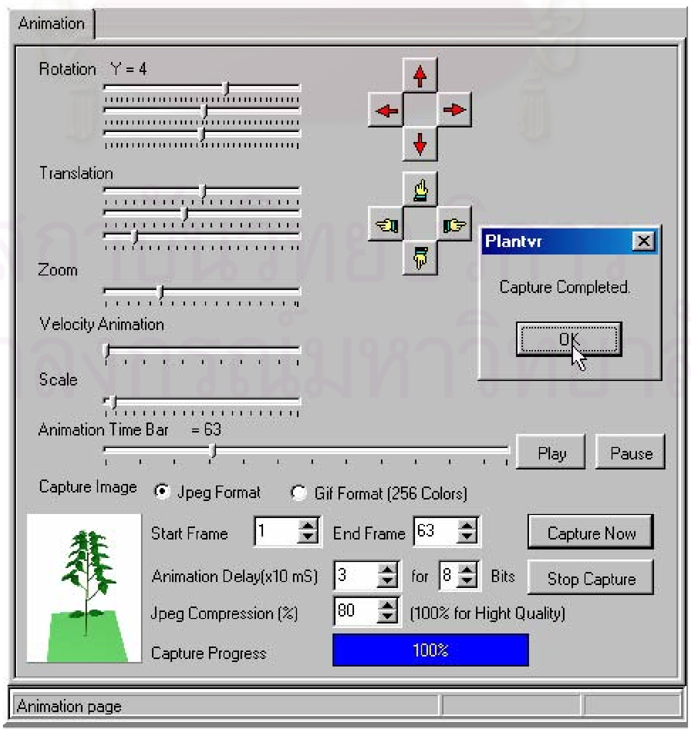


Figure C-45: Capture complete.

In the folder, the system creates the new folder name "Soybean Capture" and shows the list of animation frames picture1.jpg to picture63.jpg for each development time as Figure C-46.

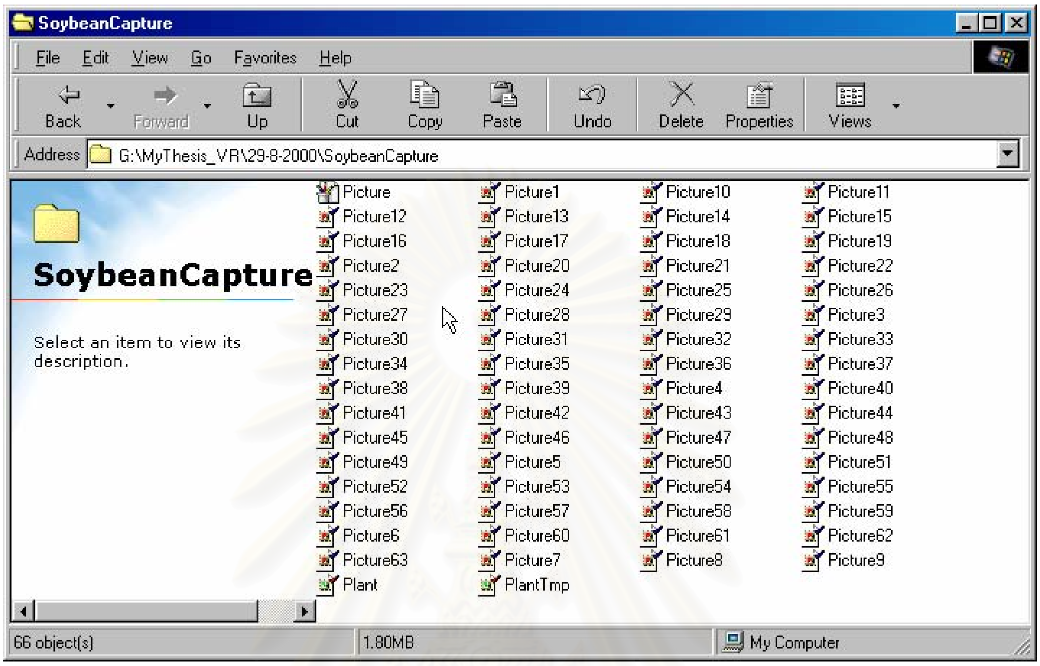


Figure C-46: The list of animation frames and the animation file.

The file picture.bmp is the last frame at the time 63. The plant.gif file is the animation file is available show on the web browser, and the temporary file planttmp.gif for the backup of the animation file in the case of captured failure. The Figure C-47 shows the plant model after completed capturing.

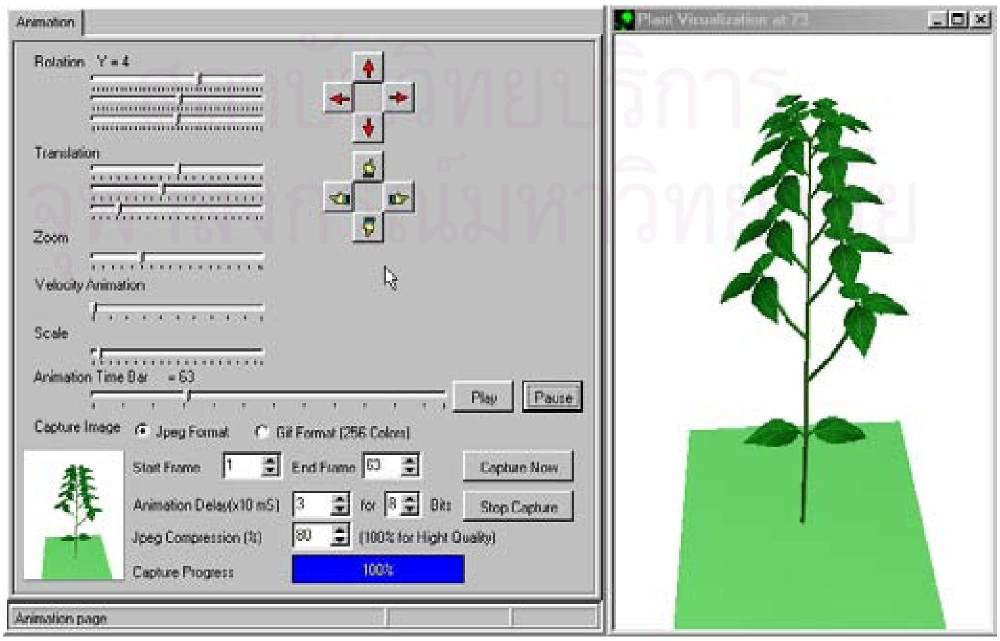


Figure C-47: The plant model after capturing complete.

7. Create a new plant

To create a new plant model, select the text editor page to copy the L-system code as Figure C-48, and paste to the L-system editor in the L-system page as Figure C-49. In the other way, type the new L-system code in the L-system editor.

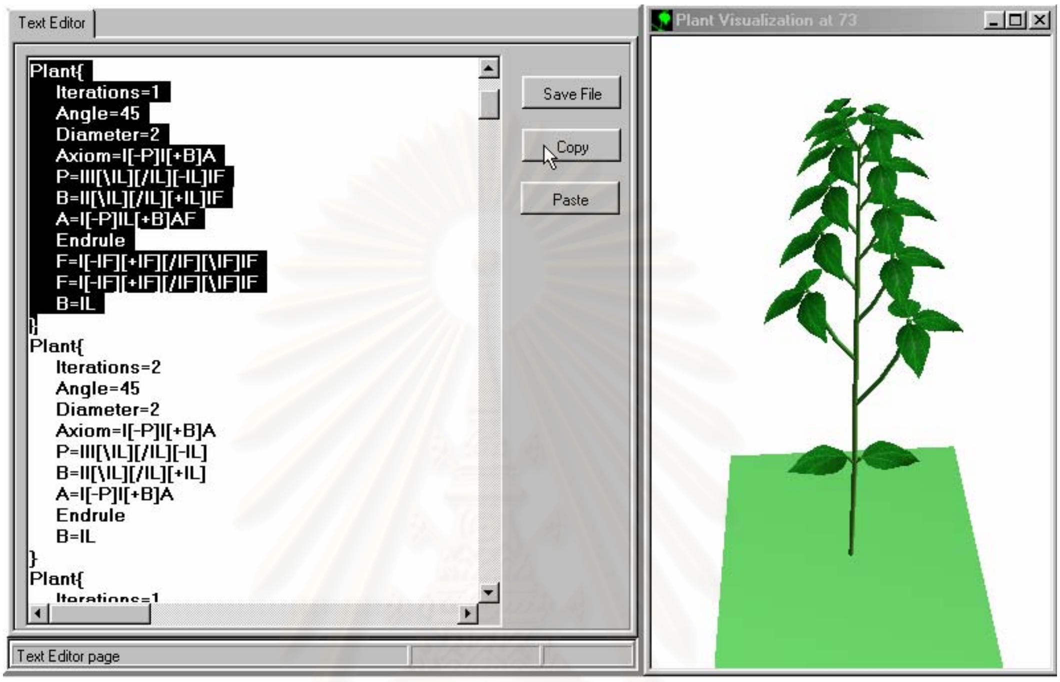


Figure C-48: Copy a new plant prototype to create the new plant.

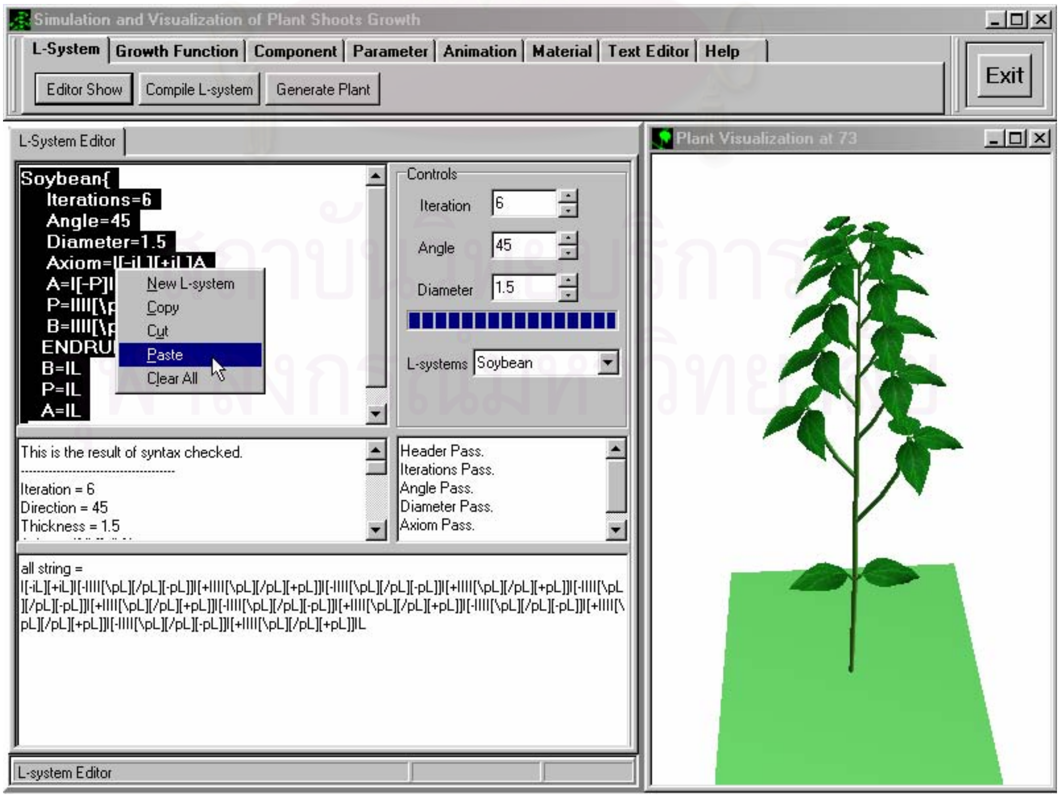


Figure C-49: Select all and paste to the soybean prototype.

Recompile and regenerate the plant, the old plant will active follow the diameter of new L-system code as Figure C-50.

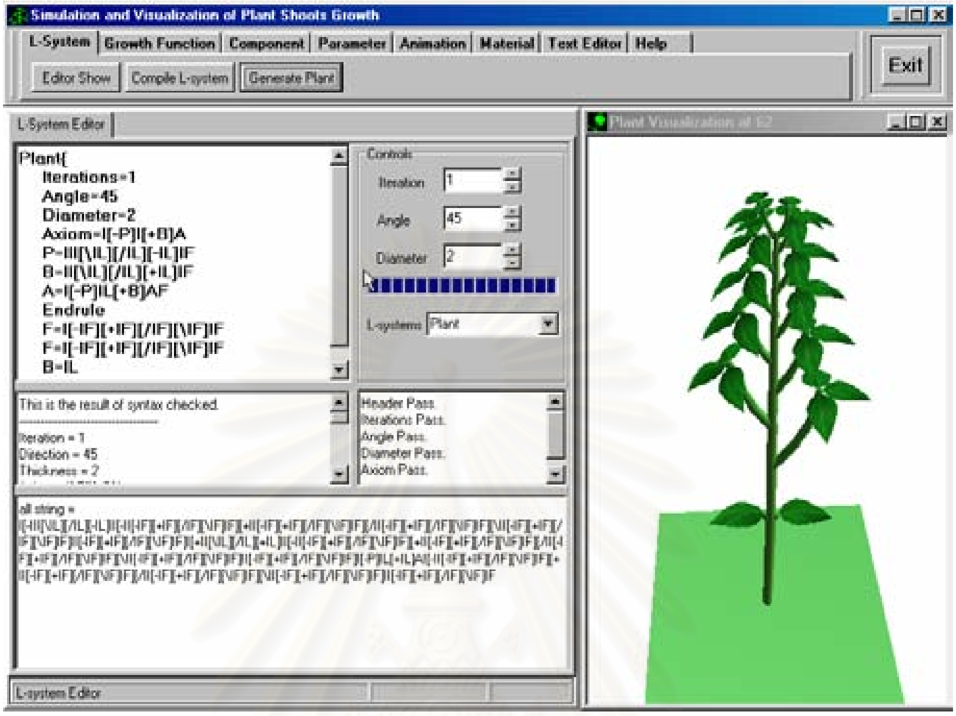


Figure C-50: Compile and generate the plant prototype.

The new plant model will be shown as Figure C-51 after the "Animation Show" is activated.

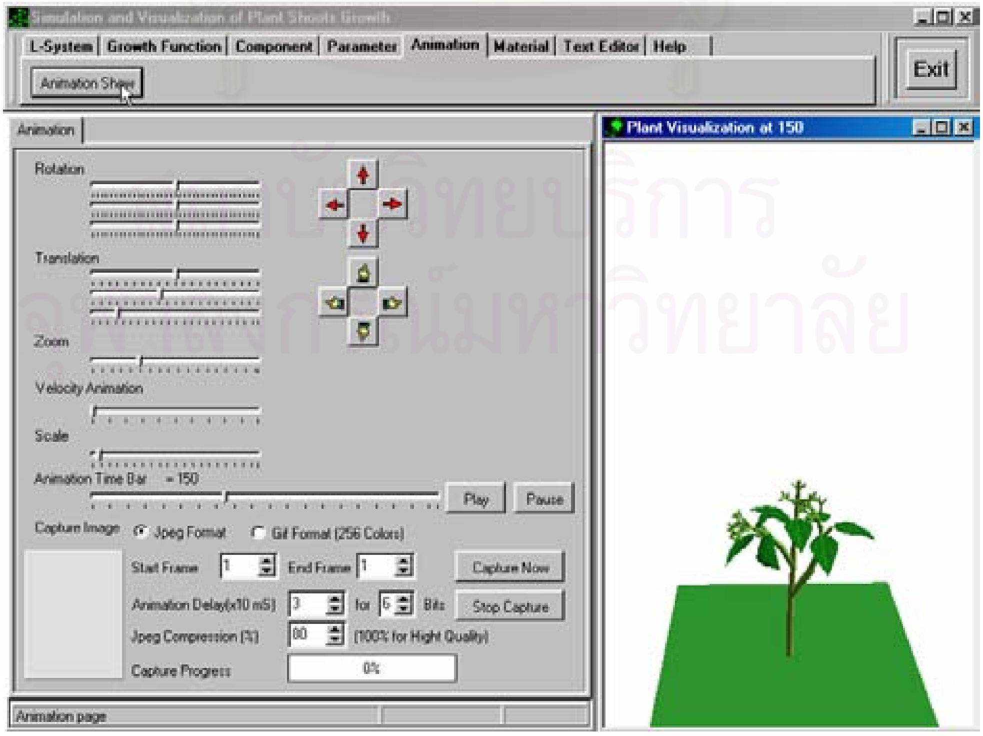


Figure C-51: The result after generating a new plant.

In the L-system string, there are the symbol "*F*" for flower component, the user must select the flower of plant such as select the petal of violet 2 as Figure C-52, but the flower are not shown.

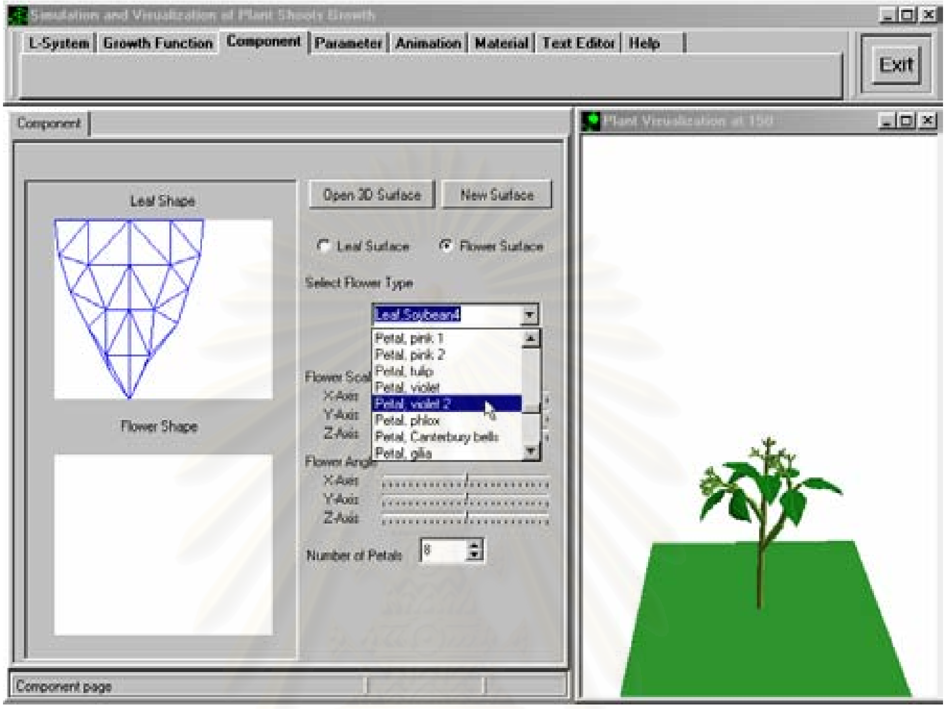


Figure C-52: Select the flower from the library.

To check the flower growth function, read data and fit the curve of flower, the flowers are shown in Figure C-53.

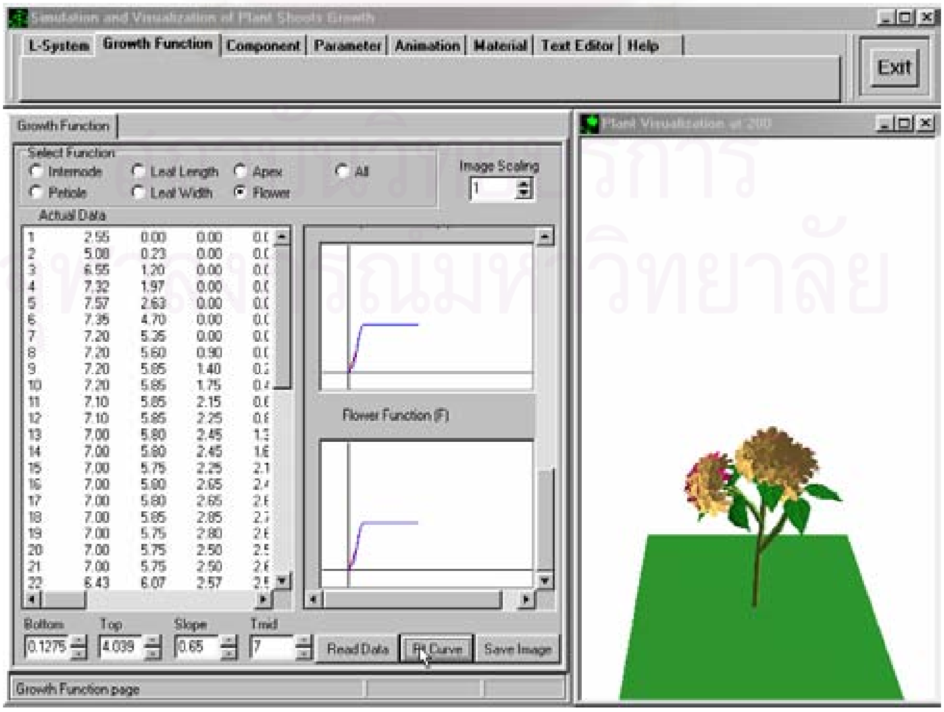


Figure C-53: Set the flower growth function using internode data.

To reduce the flower size, adjust the flower size as Figure C-54.

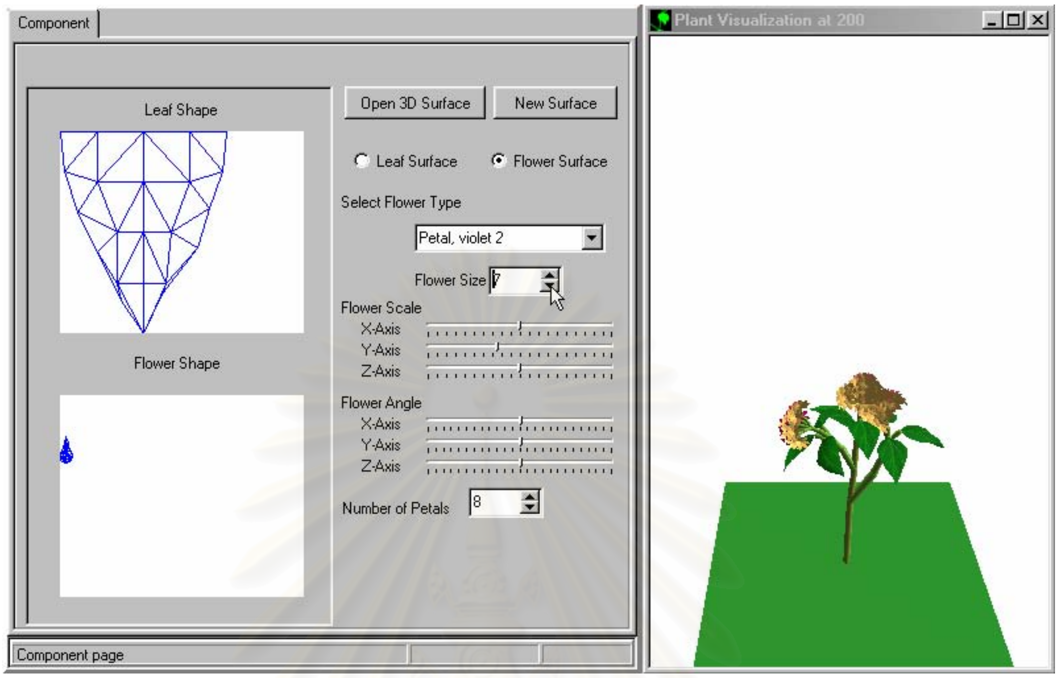


Figure C-54: Adjust the size of flower.

To set the appropriated plant, adjust the parameter such as the petiole length, the node diameter. The result will be shown as Figure C-55.

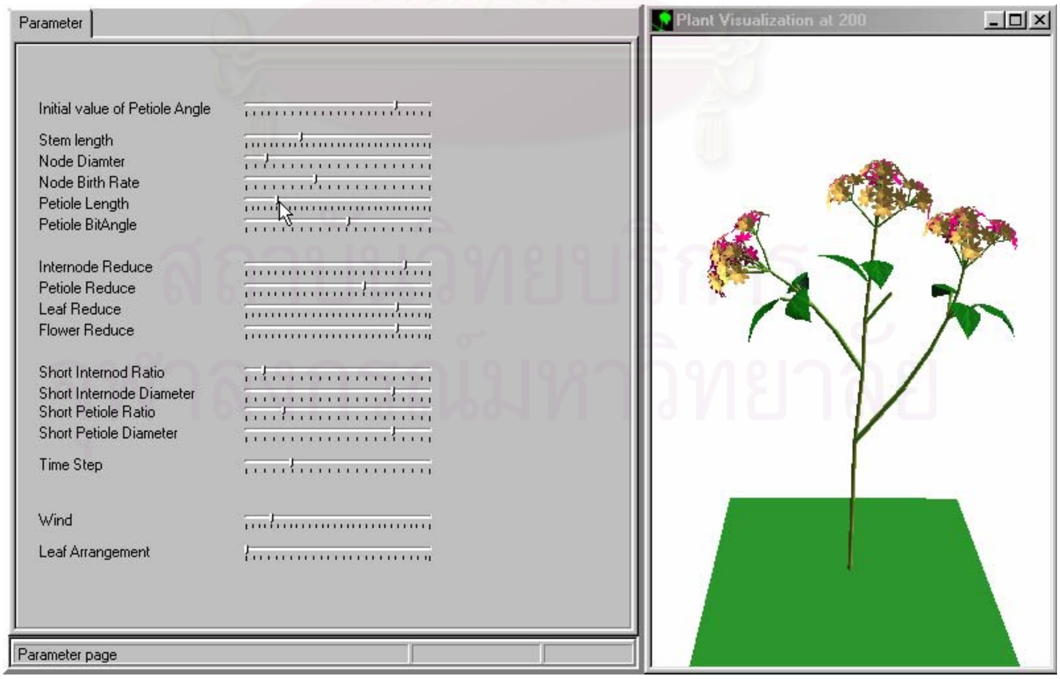


Figure C-55: Adjust the appropriated parameter.

To zoom in or translate the plant model to the appropriated perspective view, set the appropriated parameter in the animation page as Figure C-56.

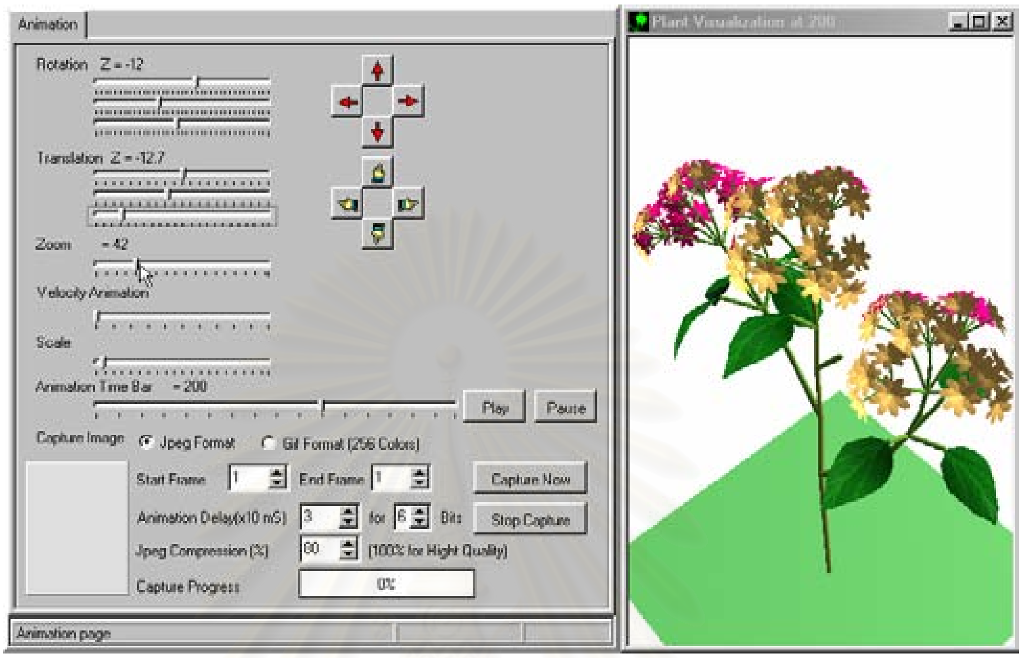


Figure C-56: Adjust the appropriated perspective view.

To change the leaf shape, select the component page and select the leaf type as Figure C-57 such as “leaf long serrated”.

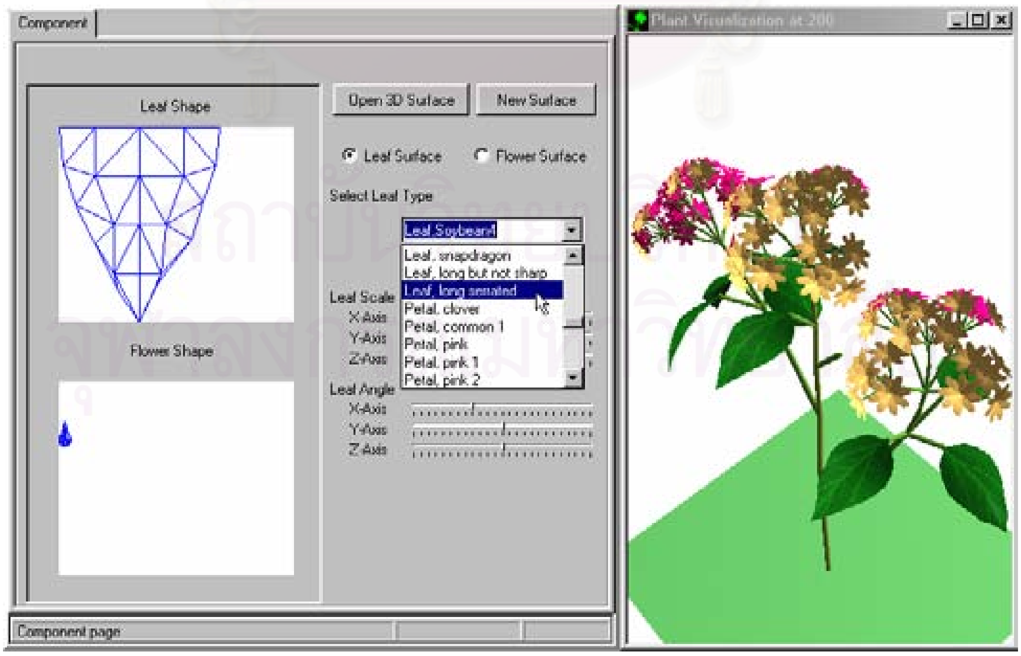


Figure C-57: Change the leaf shape of plant.

The new shape of leaf is shown in Figure C-58.

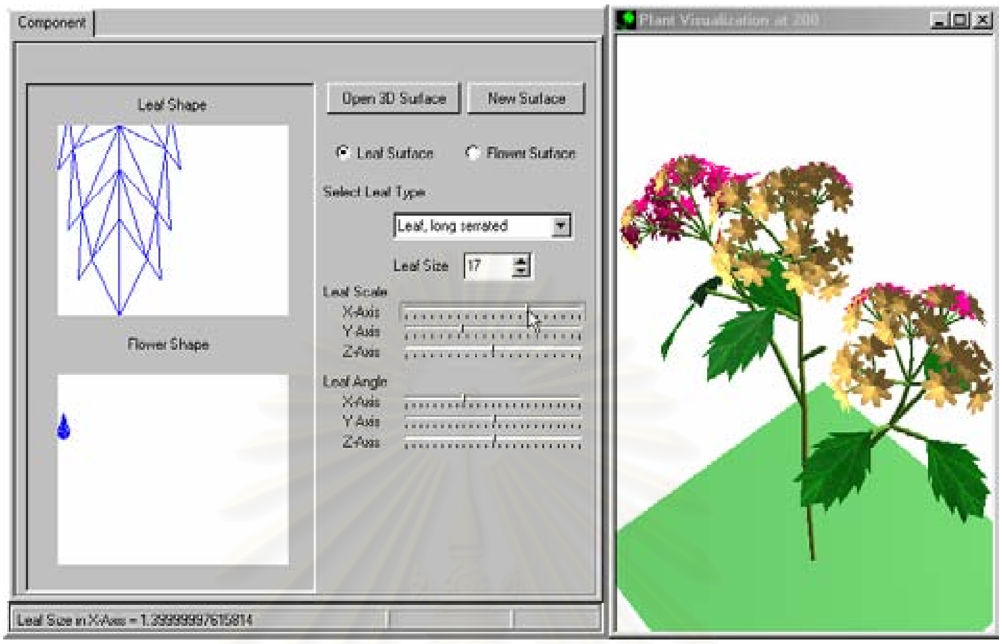


Figure C-58: The result after selecting the new shape of leaf.

To edit the new texture of leaf, click the texture image in the material page, and select the texture file as Figure C-59.

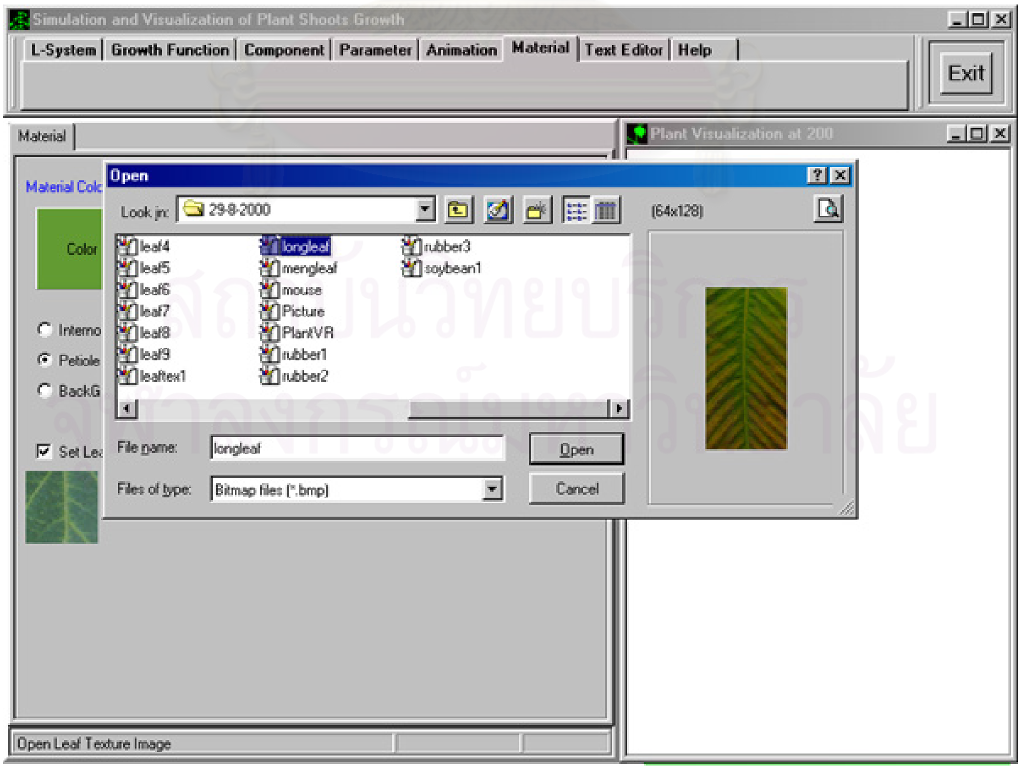


Figure C-59: Change the new texture.

The leaf shape is shown in Figure C-60 with the new leaf shape and its texture.

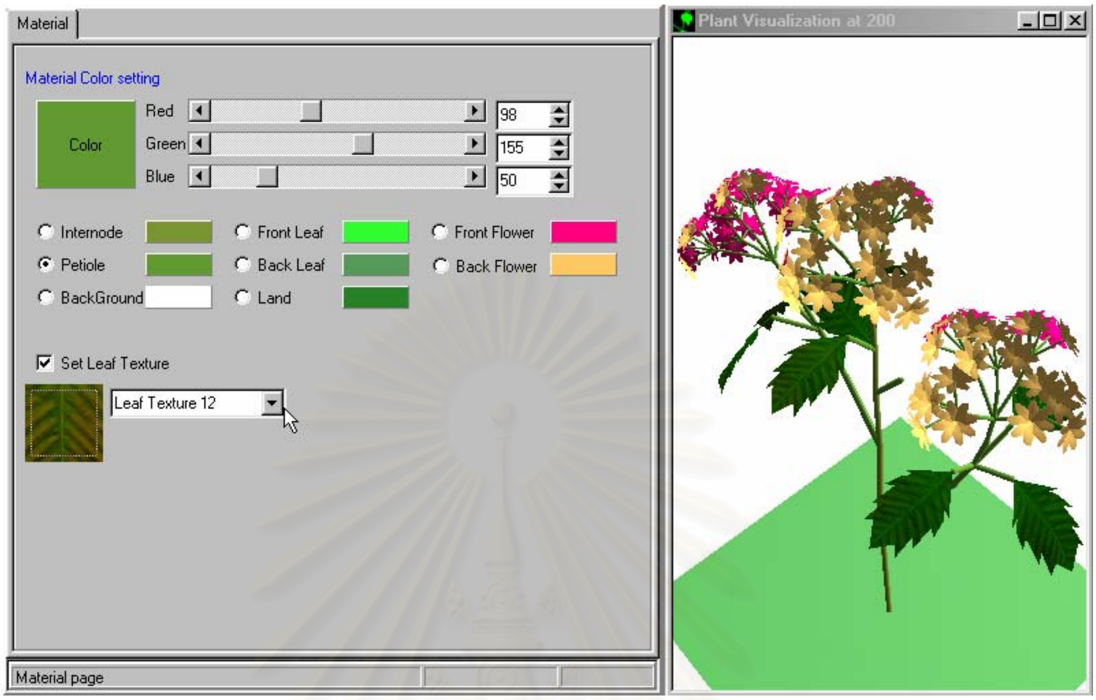


Figure C-60: The new texture result after.

To change the flower color, click the back or front of the flower radio button and adjust the preferred color as Figure C-61.

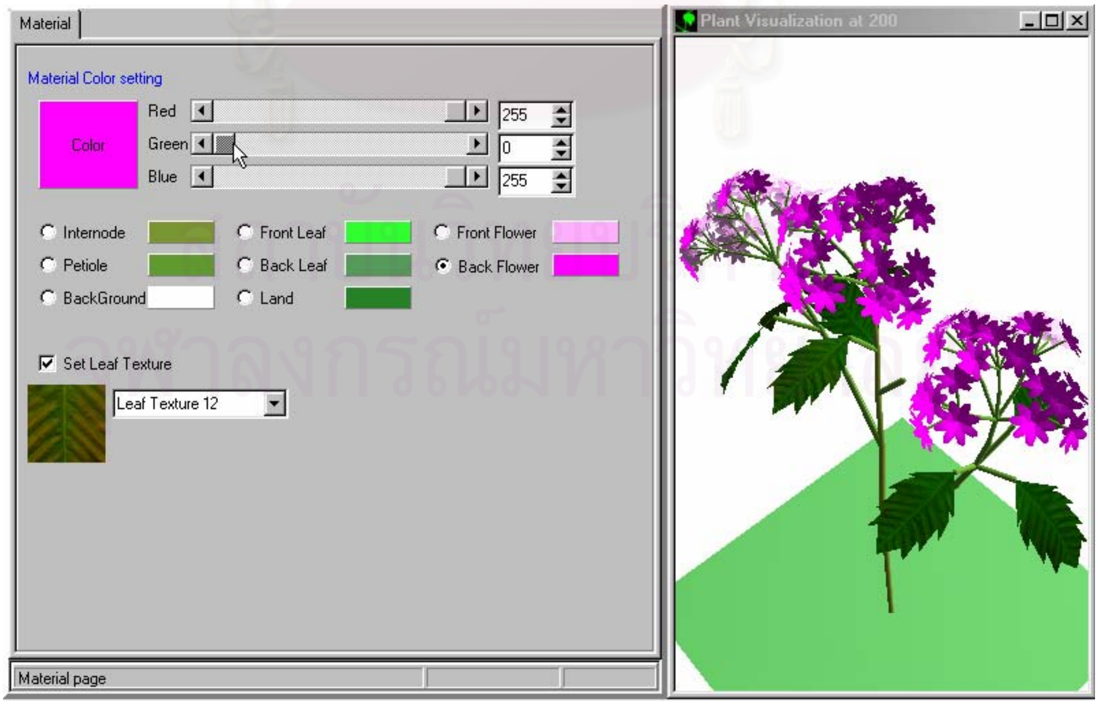


Figure C-61: Change the flower color.

To show the full screen of the plant model, click the maximize button of the visualization window or double click on the plant model. The result is shown as Figure C-62.

Figure C-62: The full screen of plant model.

To zoom in the flower, adjust the parameter on the animation page. The result will be shown as Figure C-63.

Figure C-63: Zoom in the flower.

To change the petal shape of the flower, select the flower surface such as "petal, clover". The result is shown as Figure C-64.

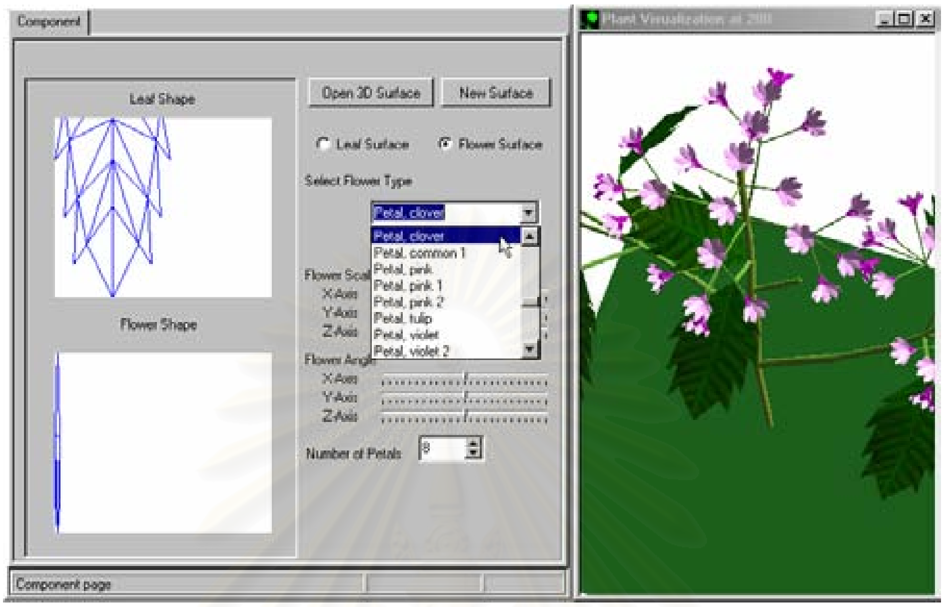


Figure C-64: Change the flower shape.

To set the flower color, select the material page and adjust the preferred color as Figure C-65.

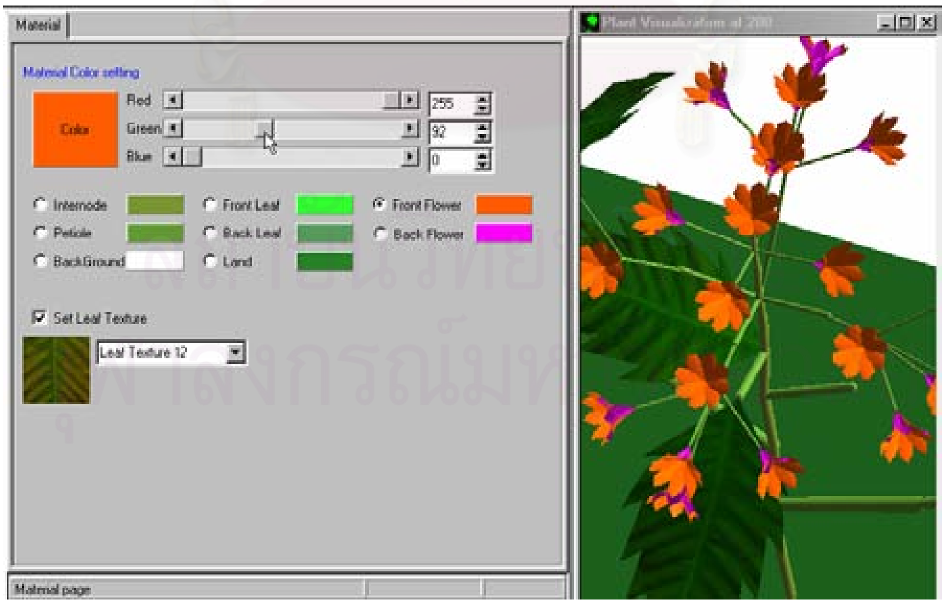


Figure C-65: Change the flower color and perspective view.

In the similar way, to change the new petal shape of the flower, select the component page, and select the petal type as Figure C-66.

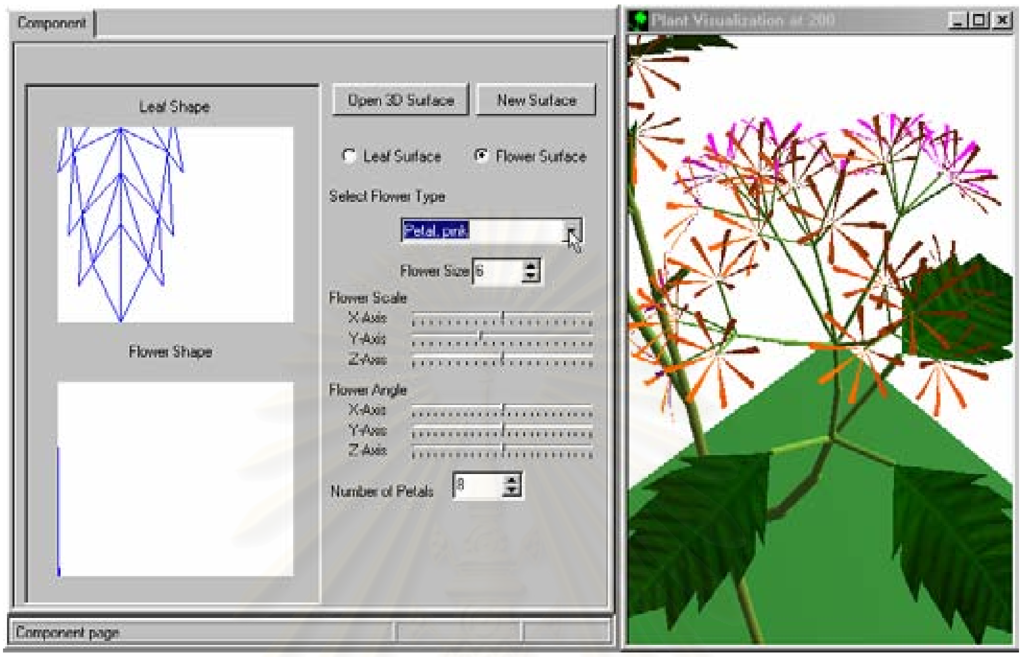


Figure C-66: Change the flower shape.

To change the number of flower petal, add the number of petal and the result will be shown as Figure C-67.

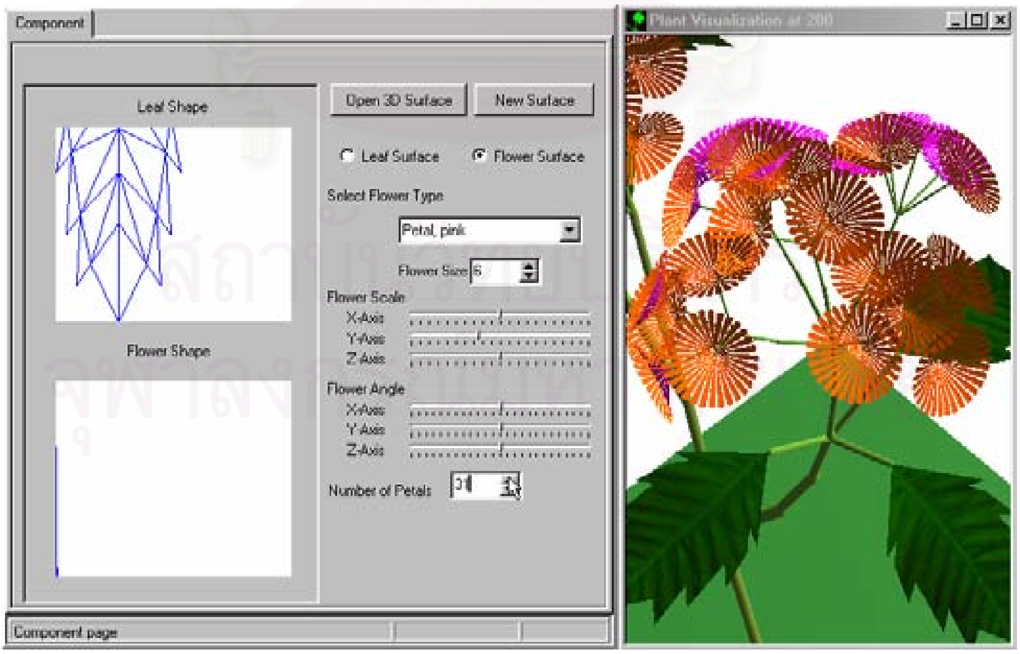


Figure C-67: Add the number of petals.

To change another petal shape, it will become the new plant in the same topology as Figure C-68, and Figure C-69.

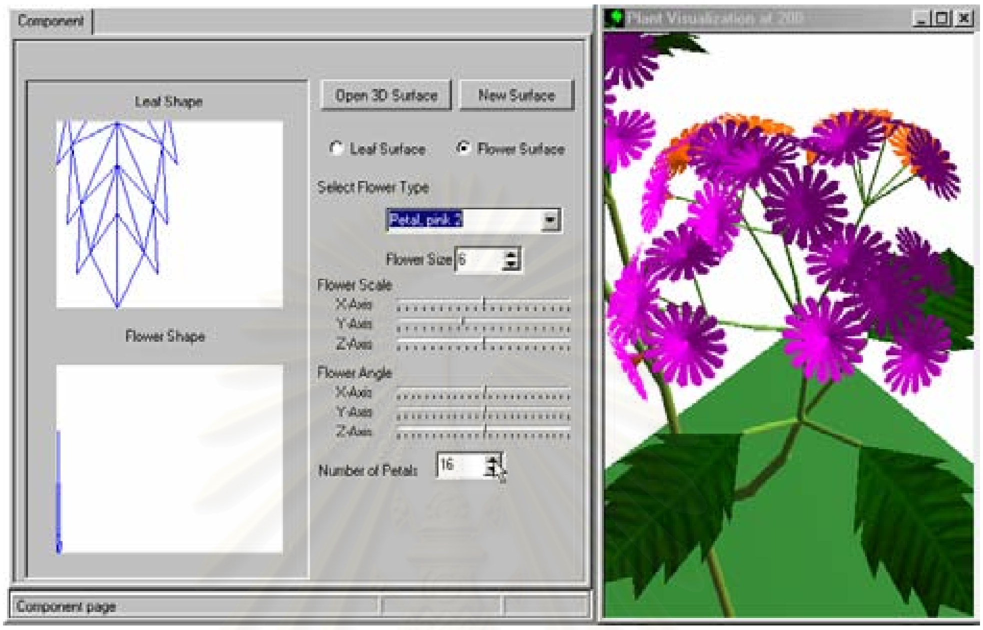


Figure C-68: Change the flower shaper and the number of petals.

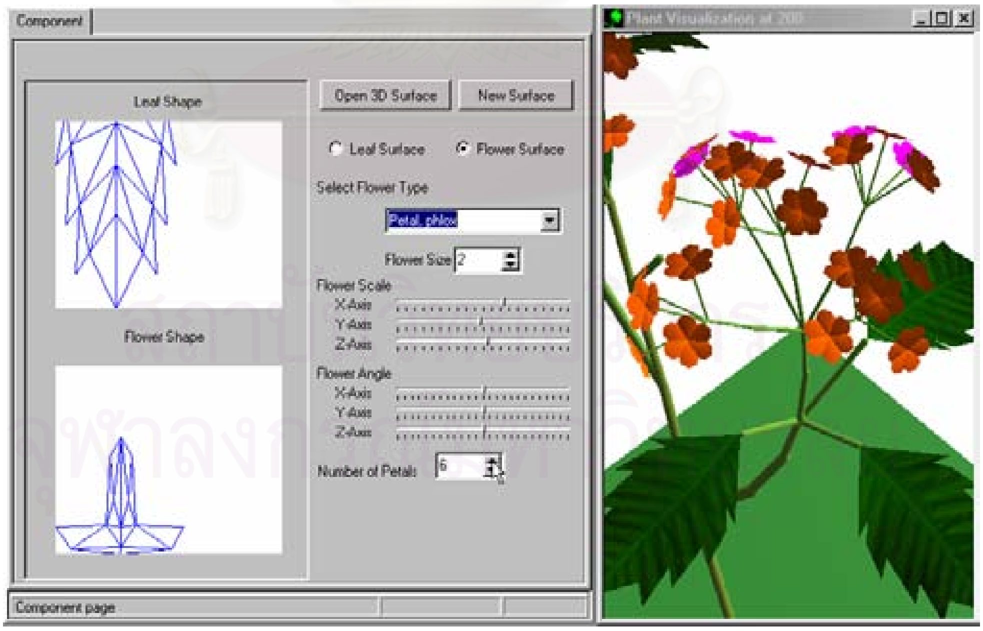


Figure C-69: Change the and six petals.

To show the new flower and the leaf color, Figure C-70 shows the new plant model.

Figure C-70: Change the flower and the leaf color.

To exit the software, click the “Exit” button on the main menu as Figure C-71.

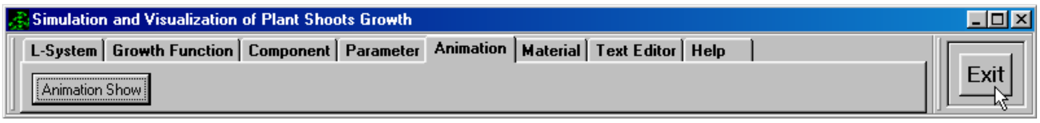


Figure C-71: Exit the program.

The leaf designer (LeafDS) software

The LeafDS software is used to design the leaf surface in three-dimensional space. To use the LeafDS software, double click the icon.

LeafDS.exe

Figure C-72: The leafDS icon.

The main program is shown in Figure C-73. The grid 5x3 points is initialized. There are six sides of leaf, there are left side, right side, top side, bottom side, front side, and back side.

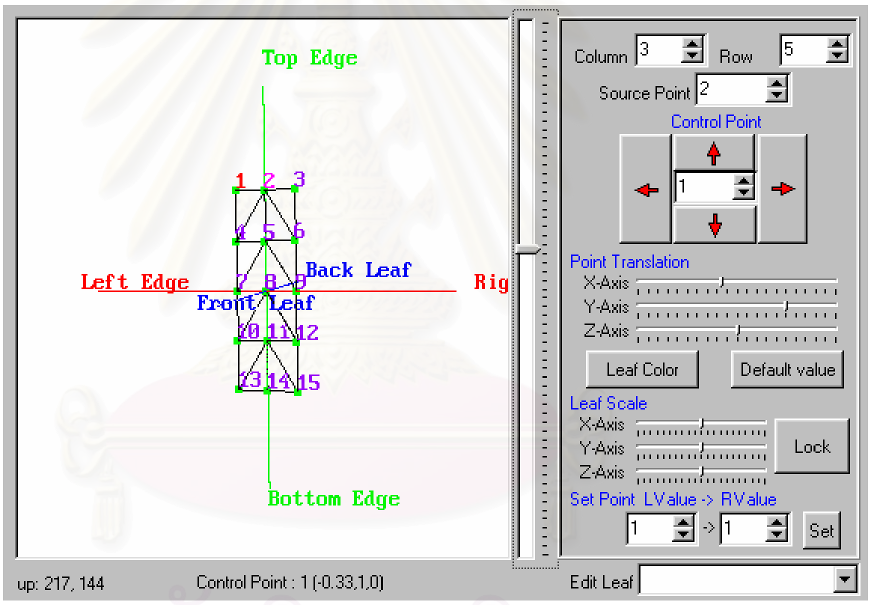


Figure C-73: The main program of LeafDS.

The control panel of the LeafDS consists is shown in Figure C-74. There are the column grid, the row grid, the source point, the control point button, the point translation, the leaf scaling, the setting point operation, the leaf color button, the default value button, the lock button, the set button, the edit leaf combo box. All component is described as follows:

1. The column grid and row grid are used to set the number of leaf grid. The default value is 5x3 points.
2. The source point is used to set the joint point to attach with the internode in the PlantVR software.
3. The control panel is used to select the control point that is controled by some operations. There are left, right, up, and down direction.

4. The point translation consists of translation on X, Y, Z axis. They are used to set the coordination of the leaf control point.
5. The leaf scaling is used to resize the leaf grid on X, Y, Z axis.
6. The setting point operation is used to set a point coordination similar to another point coordination.
7. The leaf color button is used to set the color of leaf surface.
8. The default value button is used to set the default value of control point.
9. The lock button is used to lock the grid size before designing.
10. The set button is used to set the the setting point operation.
11. The edit leaf combo box is used to select the surface from the leaf library.

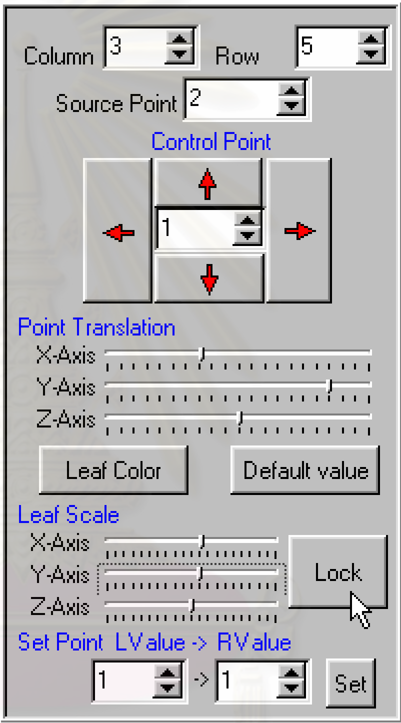


Figure C-74: The control panel of leaf designer.

To show the XYZ axis, right click on the the output view. Select the popup menu “Hide Leaf Axis” as Figure C-75. The result is shown in Figure C-76.

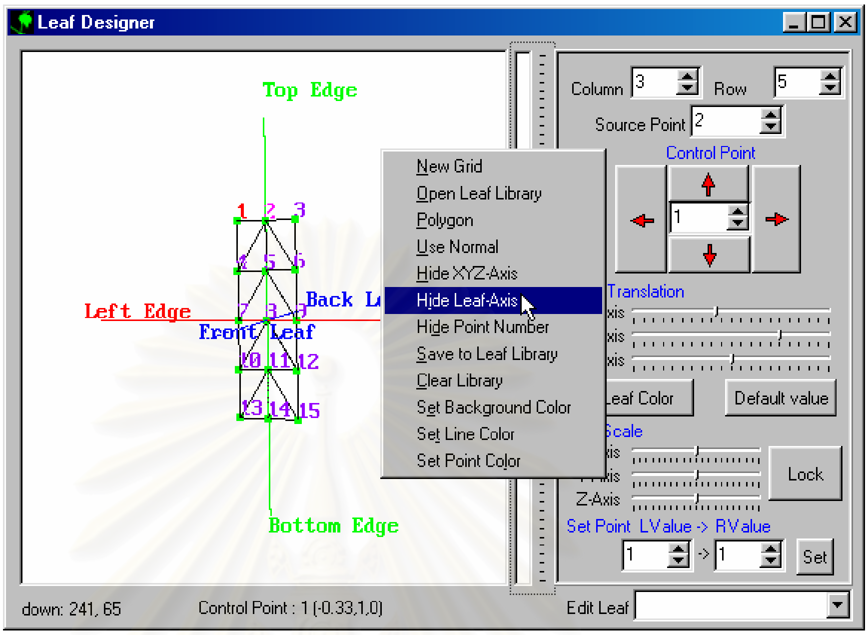


Figure C-75: Hide the leaf axis to XYZ axis.

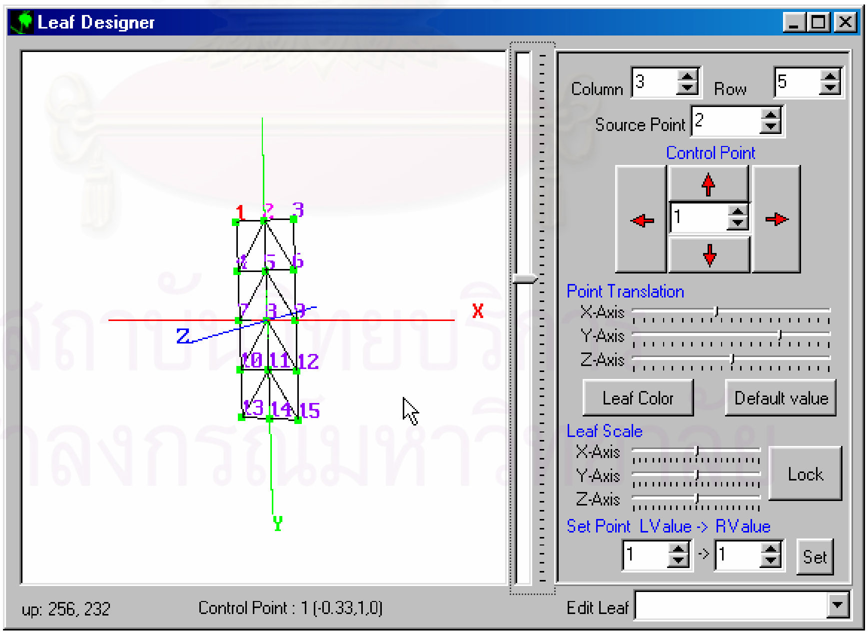


Figure C-76: The result after change the axis.

To zoom in and out the leaf structure, click the vertical trackbar as Figure C-77.

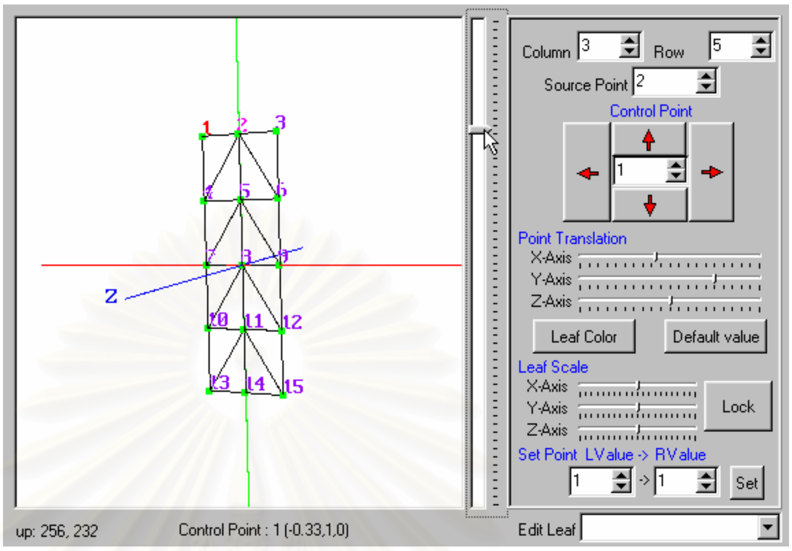


Figure C-77: Zoom in and out.

To set the grid size of the leaf, adjust the horizontal trackbar at “Leaf Scale”, adjust the X-axis to resize along X-axis as Figure C-78, adjust the Y-axis to resize along Y-axis as Figure C-79, adjust the Z-axis to resize along Z-axis. Then click “Lock” button to lock the grid size as Figure C-80.

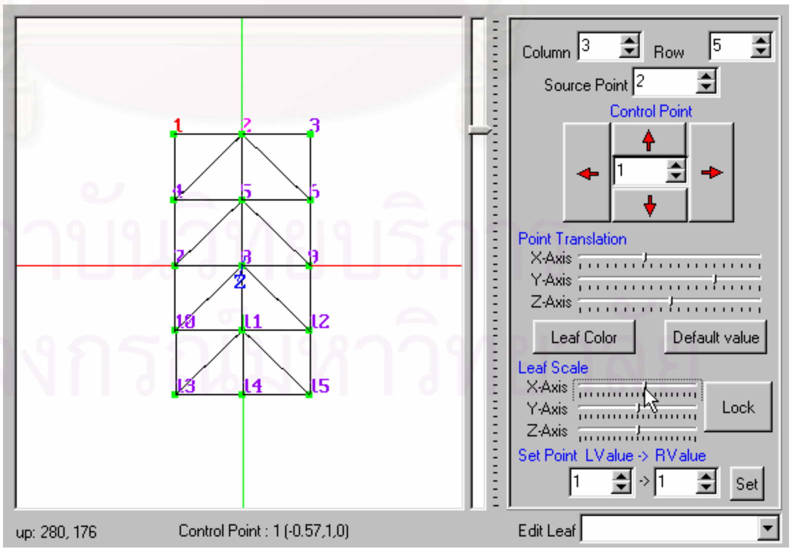


Figure C-78: Adjust X-axis to resize the leaf.

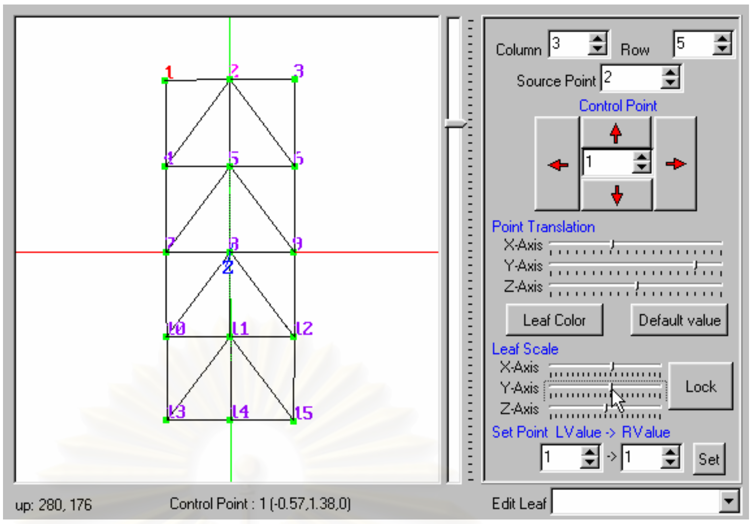


Figure C-79: Adjust Y-axis to resize the leaf.

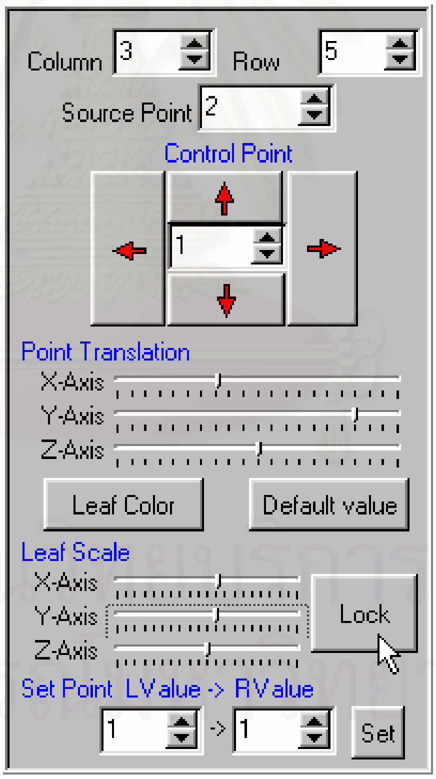


Figure C-80: Lock the grid resize.

To set the grid to the leaf shape, first, set the first point and third point similar to second point as Figure C-81 and Figure C-83. The results are shown in Figure C-82, and Figure C-84, respectively.

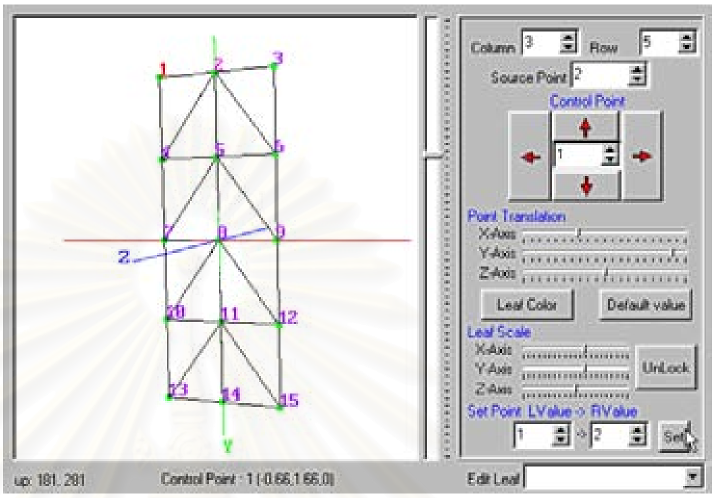


Figure C-81: Set the first point to second point.

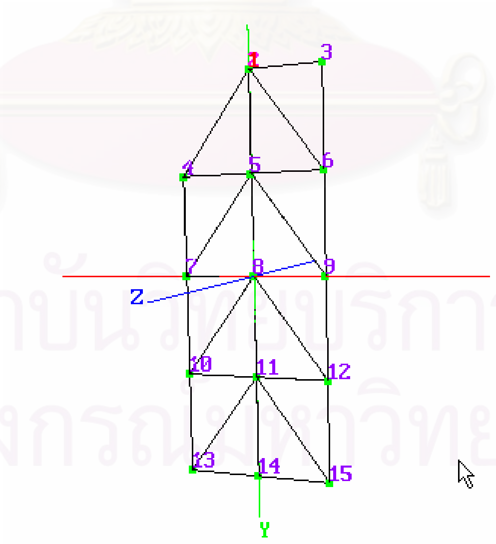


Figure C-82: The result after setting the first point to second point.

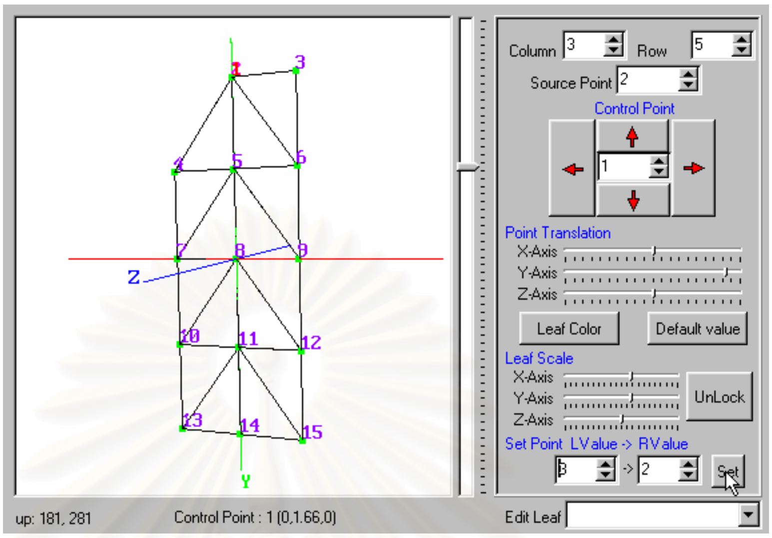


Figure C-83: Set the third point to second point.

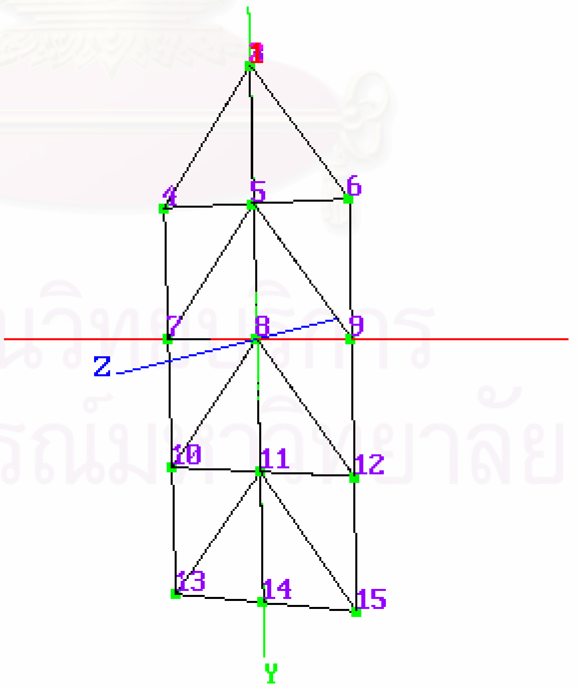


Figure C-84: The result after setting the third point to second point.

In the similar way, set the 13th point and 15th point similar to the 14th point as Figure C-85. The result is shown in Figure C-86.

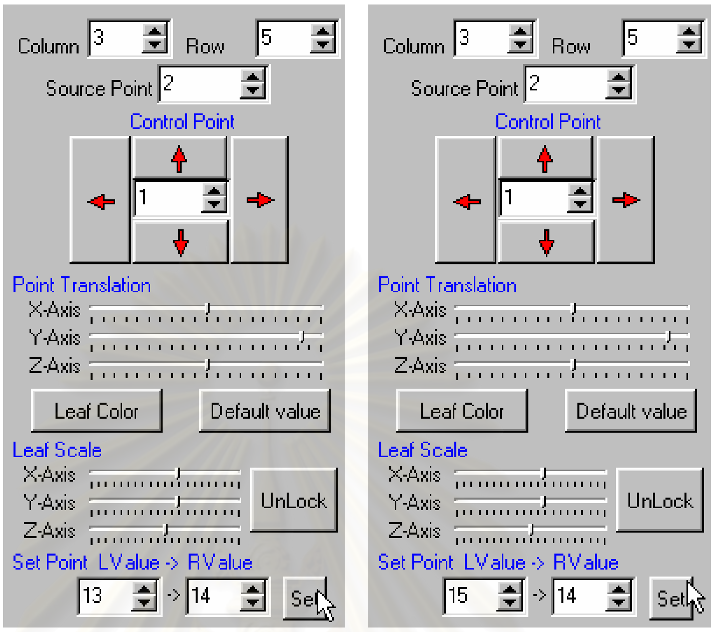


Figure C-85: Set the 13th and 15th point to 14th point.

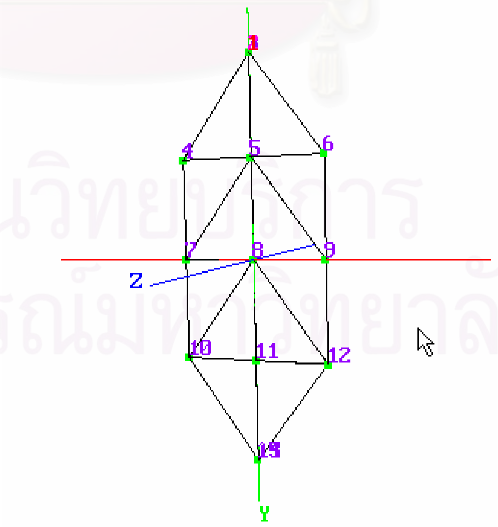


Figure C-86: The result after setting the 13th point and 15th point to 14th point.

To move the seventh point to the left direction, select the control point as Figure C-87, and adjust the “Point Translation” on X-axis trackbar to the leaf side. The result is shown in Figure C-88.

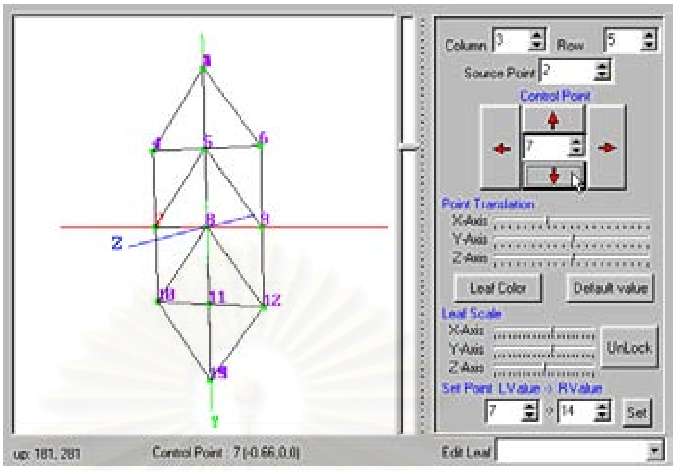


Figure C-87: Select the senventh control point.

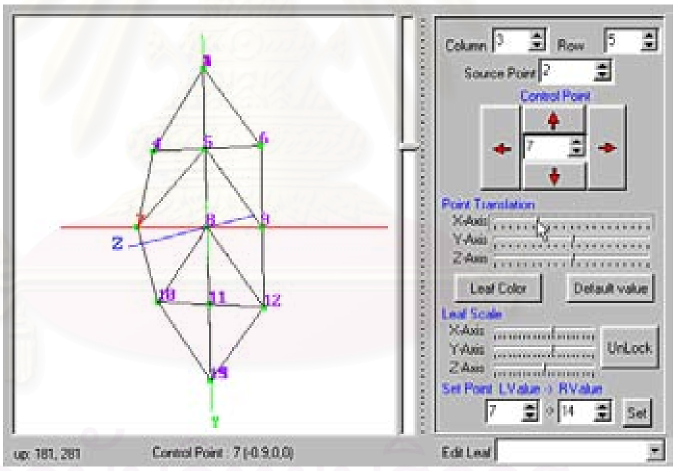


Figure C-88: Move the seventh control point.

In the same way, select the ninth point and move to the right size on the X-axis as Figure C-89.

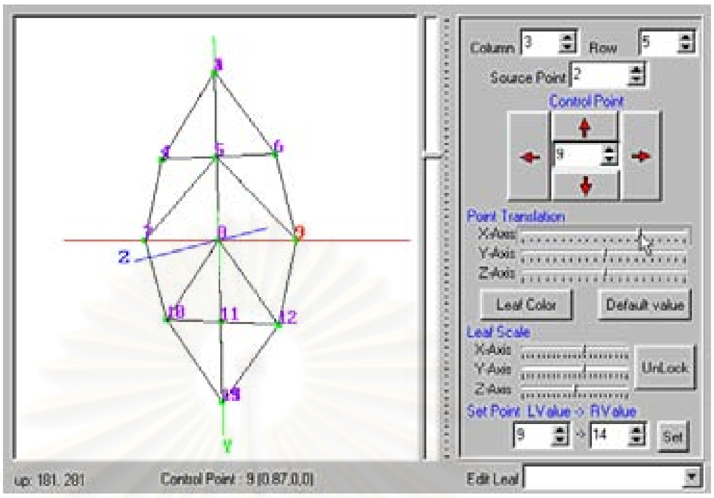


Figure C-89: Move the ninth control point.

To change the perspective view, move and drag the mouse over the output view as Figure C-90.

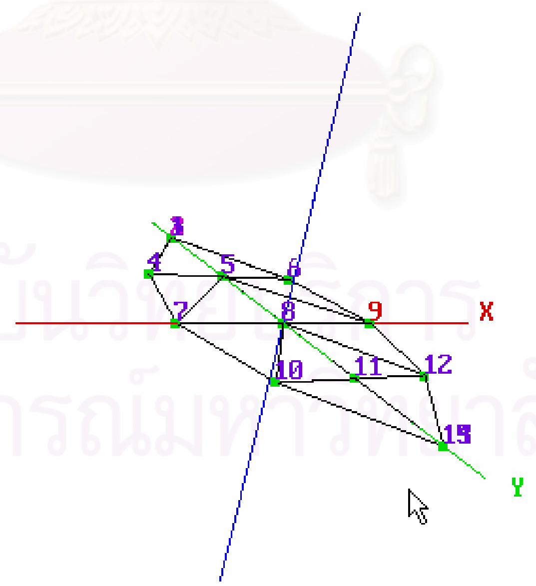


Figure C-90: New perspective view.

The leaf is now flat on the Z-axis, to set the mid rib upward, select the fifth control point and move upward as Figure C-91.

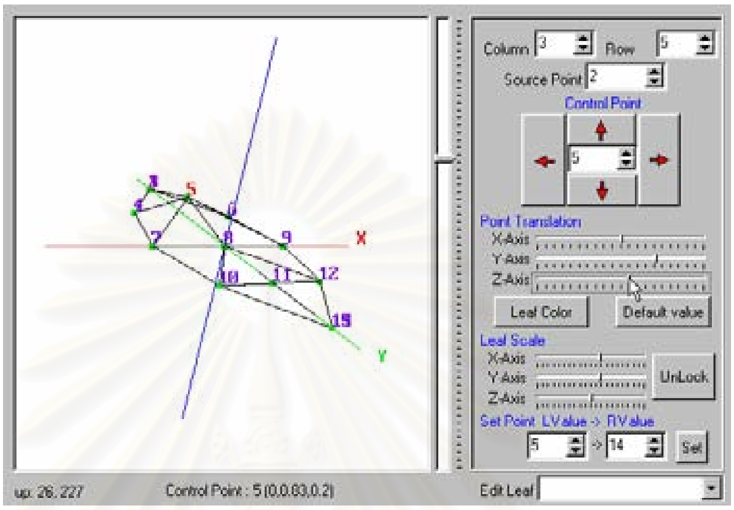


Figure C-91: Move the fifth point upward.

To move the eighth control point, select the eighth control point and adjust the trackbar to move to the appropriated position as Figure C-92.

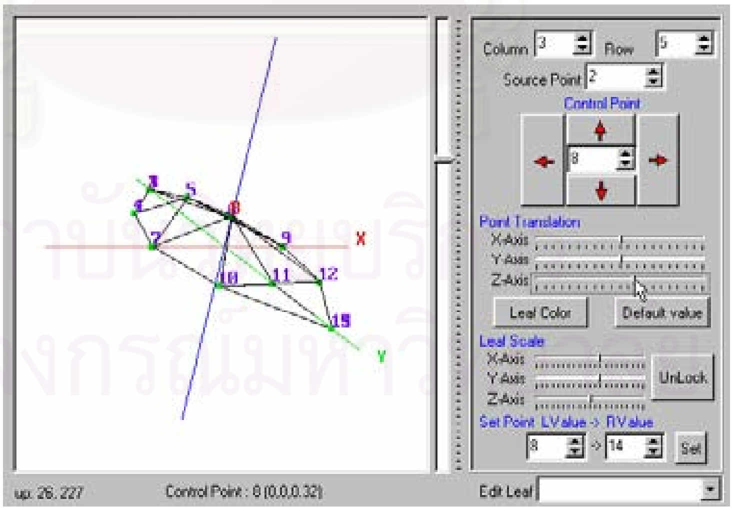


Figure C-92: Move the eighth point upward.

In the similar way, to move the 11th control point upward, select the 11th control point and adjust the trackbar on Z-axis to the right size for upwarding as Figure Figure C-93.

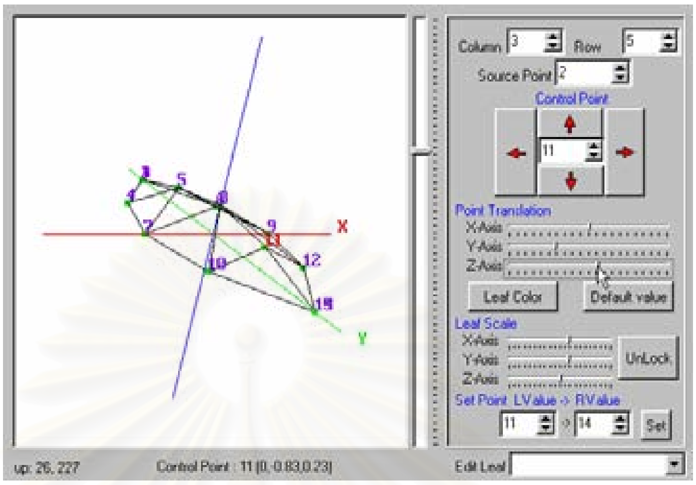


Figure C-93: Move the 11th point upward.

The result of the leaf shape is shown in Figure C-94. The fourth, sixth, seventh, ninth, 10th and 12th control points are now on the same Z value or the same plane. Setting to the perfect leaf, move these control point upward.

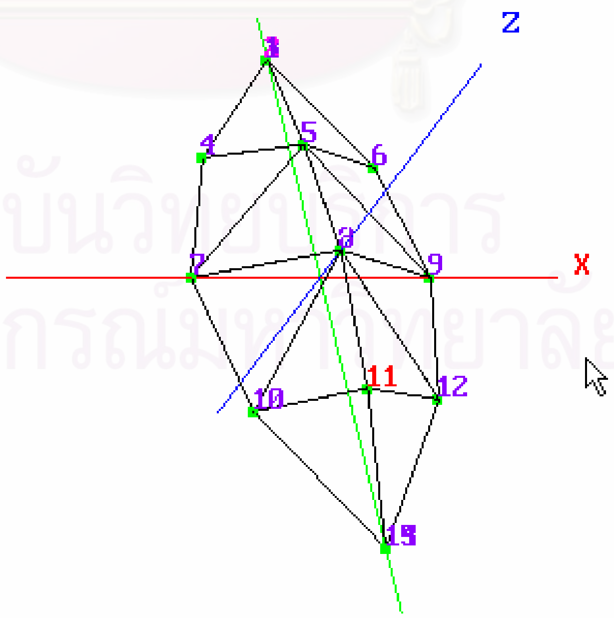


Figure C-94: The leaf shape.

The XZ-plane of leaf shape is shown in Figure C-95 and move the fourth control point upward.

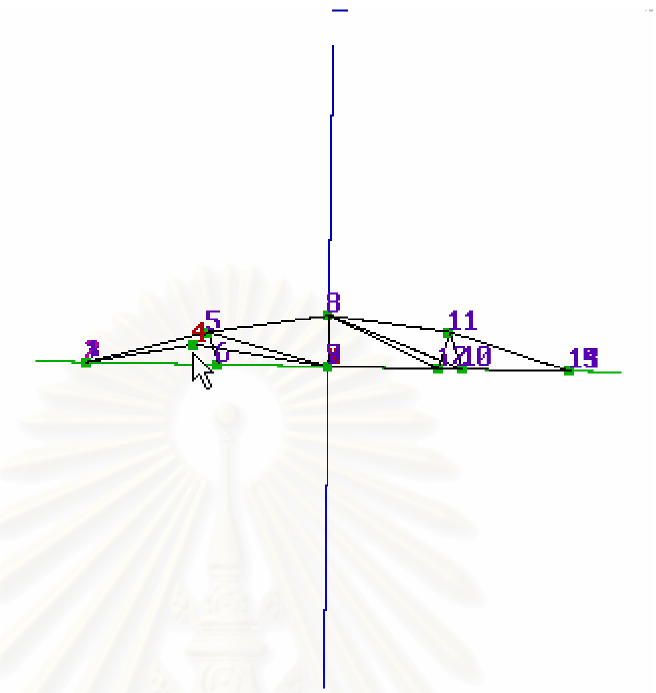


Figure C-95: The leaf shape.on XZ-plane.

Select the sixth control point and move the sixth control point to upward as Figure C-96.

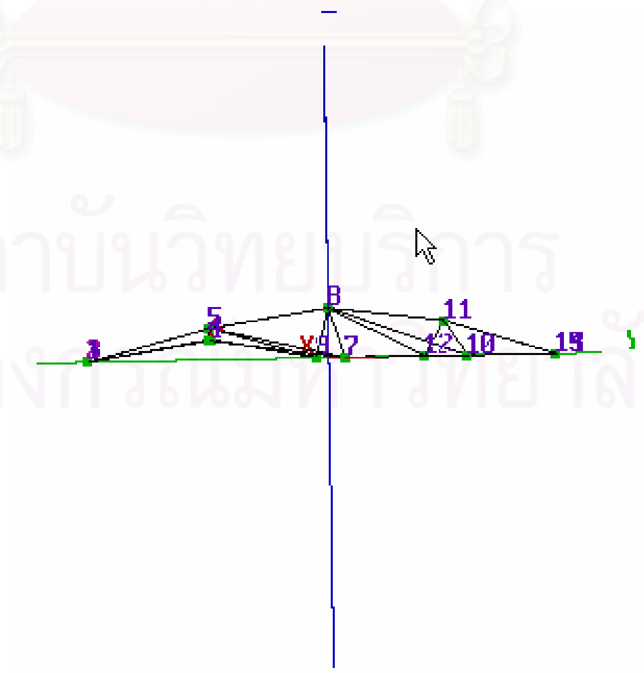


Figure C-96: Move the sixth point upwarding.

Select the seventh control point and move upward, and the same way, select the ninth control point and move upward as Figure C-97. The result is shown in Figure C-98.

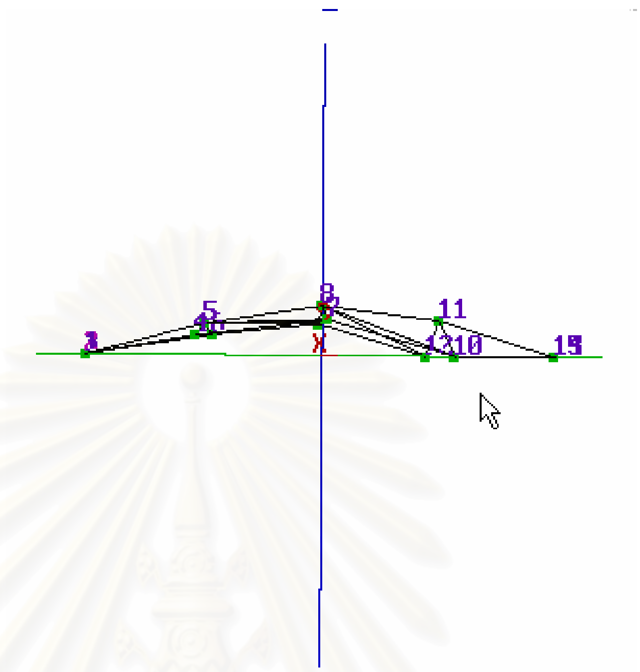


Figure C-97: Move the seventh and ninth point upward.

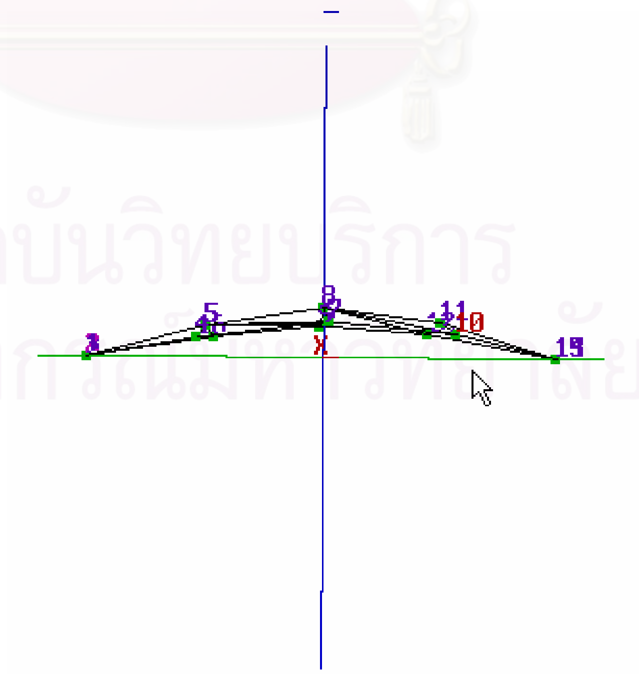


Figure C-98: The result after move upward.

The new perspective view is shown in Figure C-99.

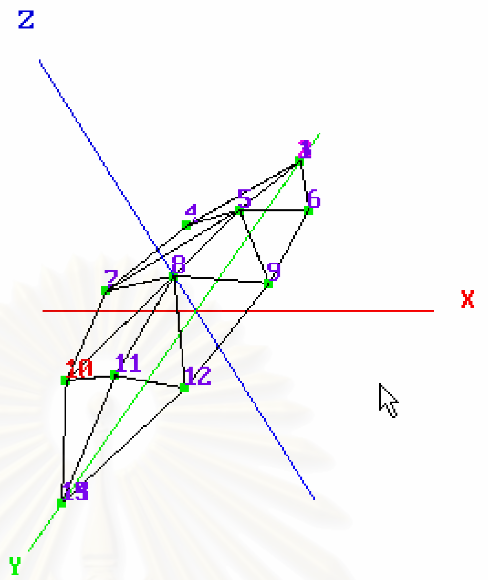


Figure C-99: The new perspective view.

To show the polygon on the leaf surface, right click on the output view and select the popup menu "Polygon" as Figure C-100. The result of the polygon is shown in Figure C-101.

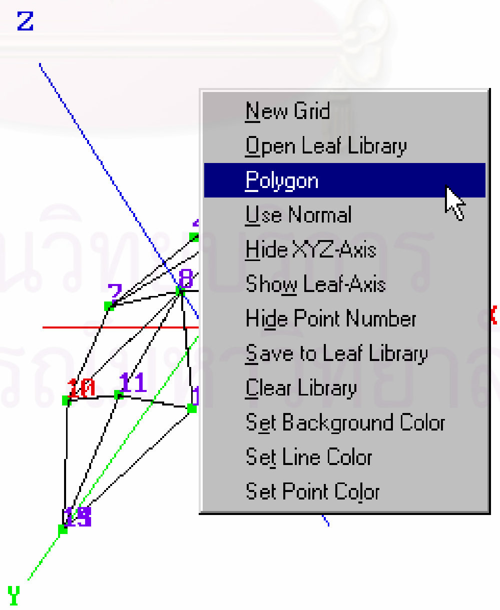


Figure C-100: Set the polygon to the leaf surface.

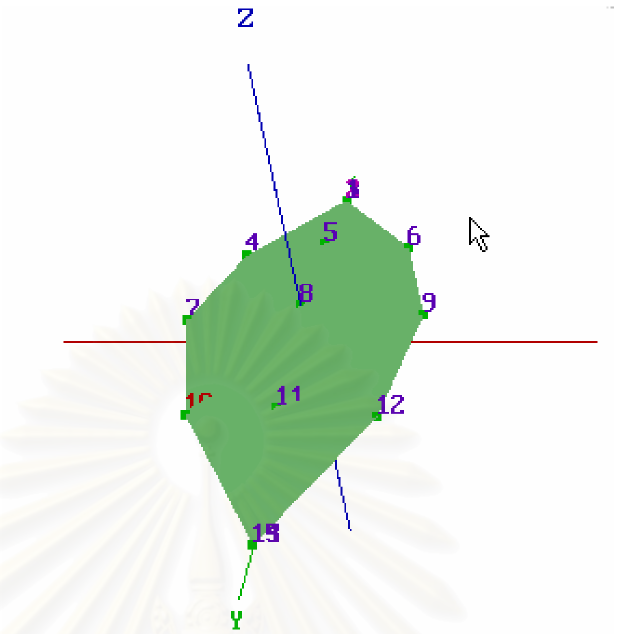


Figure C-101: The flat shading leaf using triangular rendering.

The new perspective of leaf shape is shown in Figure C-102.

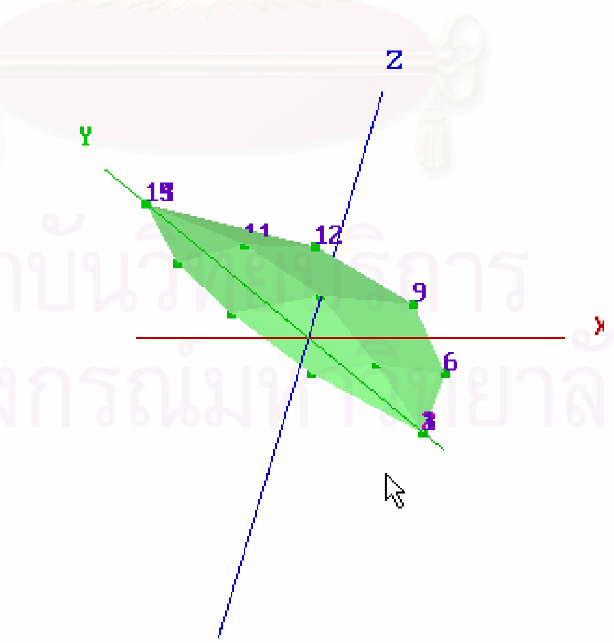


Figure C-102: The new perspective of leaf.

To hide the XYZ-axes, right click on the output view and select the popup menu "Hide XYZ-axis" as Figure C-103. The result is shown in Figure C-104.

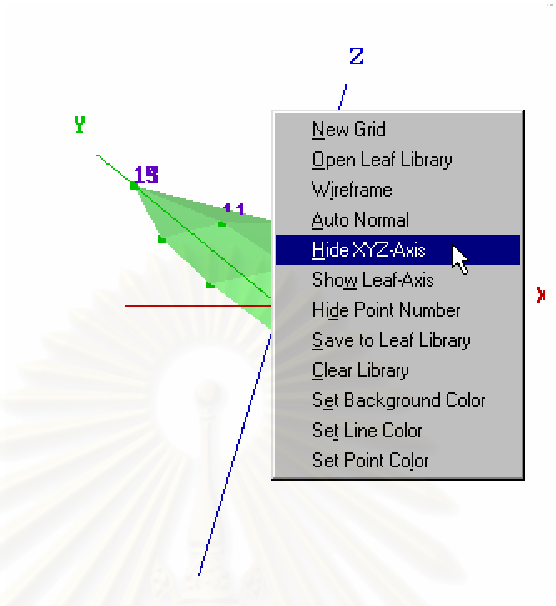


Figure C-103: Hide the XYZ-axes.

Figure C-104: The result output after hiding XYZ-axes.

To save the perfect leaf to the library file, right click on the output view and select the popup menu "Save to Leaf Library" as Figure C-105.

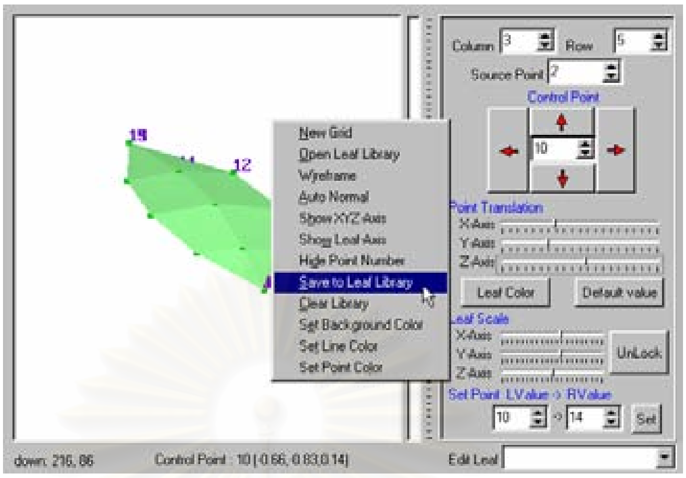


Figure C-105: Save to the leaf library.

Enter the surface type and the type name to the input box as Figure C-106, such as "Leaf, TestLeaf" for the leaf surface and the testleaf name. Enter "petal, flowerleaf" for the flower surface

Figure C-106: Input the surface type and name.

Enter the filename of the surface to the “Save As” window as Figure C-107. It is used to import to the PlantVR software.

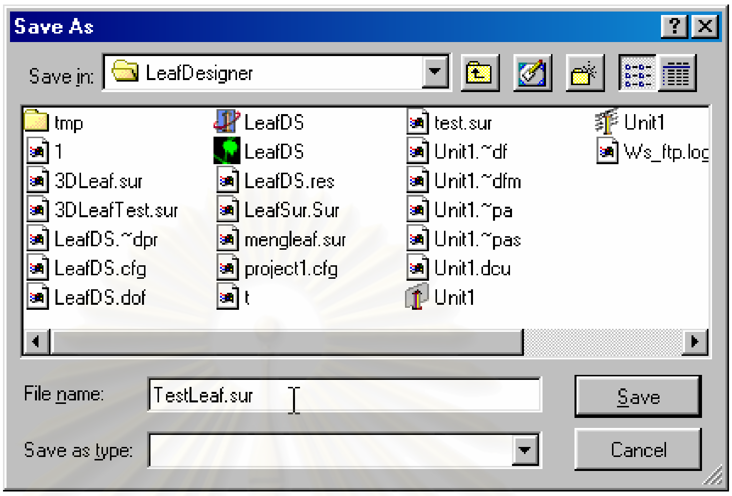


Figure C-107: Save the library name.

To create the new grid, right click on the output view and select the popup menu “New Grid” as Figure C-108.

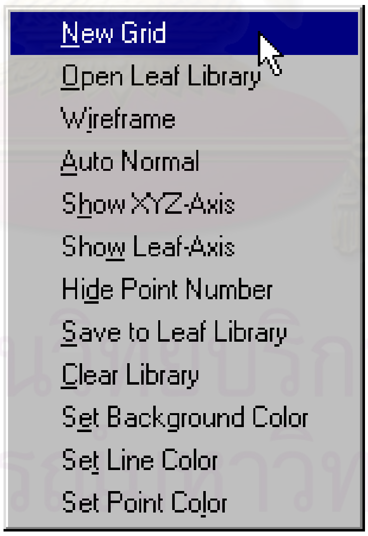


Figure C-108: The Popup Menu.

To design the complexity of leaf shape, the grid is changed as the preferred leaf. The grid is designed for 3x3 to 11x11 points, and set or adjust the control point as the previous example leaf. For example, the grid 7x5 is used in Figure C-109.

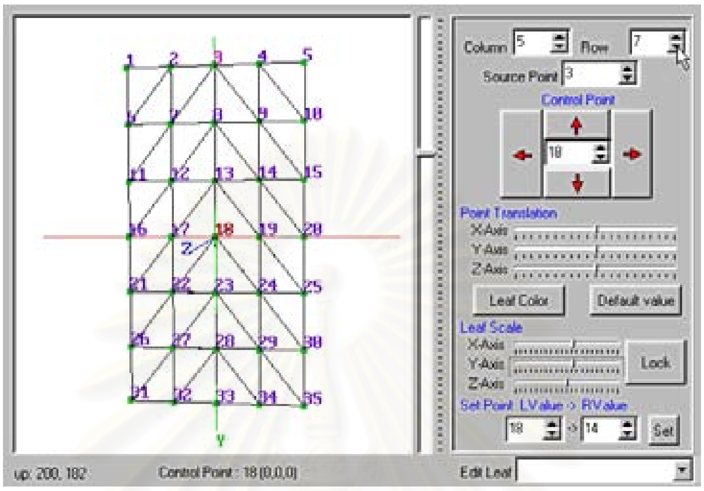


Figure C-109: The new leaf design grid.

The LeafDS is used to design the leaf and flower library, they should be stored in the same or different file.

Curriculum Vitae

Somporn Chuai-aree was born in September 30, 1974, in Nakhon sri thammarat. He received a bachelor degree in Applied Mathematics from the Department of Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University in 1996. He is now a lecturer at the Department of Mathematics and Computer Science, Faculty of Science and Technology at Prince of Songkla University, Pattani campus, Thailand.