Real-time 3D Plant Structure Modeling by L-System with Actual Measurement Parameters

Rawin Viruchpintu and Noppadon Khiripet
National Electronics and Computer Technology Center, Pathumthani 12120, Thailand.
rawin.viruchpintu@nectec.or.th, khirin@nectec.or.th

Abstract

Plant modeling is the process combining biological knowledge, mathematical formalisms and computer graphic to generate virtual plants. Recently, the most popular technique used to model plants is the Lindenmayer system or L-system. This technique presents self-similarity patterns of a tree by its 3D turtle grammar. While the resulting rendering of the traditional L-system may look like a real tree, no parameters of the modeled tree are taken into consideration. In this study, the correct parameters such as the angle between branches, the length ratio of branches and phyllotactic patterns were integrated into our customized L-system interpreter. The 3D interaction and simulation of this virtual tree were developed using java3D technology. The final 3D model can also be exported to other sophisticated 3D modeling software via the Wave-front OBJ file format. Our approach can be easily extended and applied to other 3D modeling problems requiring more faithful results.

1. Introduction

Modeling objects with mathematic and computational tools is always a crucial step in understanding the real process and situation. In biological domain, such as plant modeling, this requires biological knowledge of plant physiology along with mathematical formalism and computer graphic. There has been an active research area in modeling plant using Lindenmayer System [1], which simplify the plant shape as a 3D object. The object is defined by a self-similarity grammar. Many existing tools nowadays can facilitate the generating of grammar to represent any particular plants. For example, L-System Generation Program (LSystem4) [2] is a 3D visualization tool which offer the enhance options for viewing 3D model such as rotation and movement control, texture mapping per layers and visible assignment of textures to layers. J. Scott Cameron has shown the wonderful 3D L-Systems application [3] to model the growth of trees. His L-system was nicely implemented by the use of Microsoft's Direct3D API to show the skeleton of the tree structure. LYNDYHOP [4] is a simple Java-Application for generating self-similar fractals in 2D graphics in step by step. The program also provides a graphical representation of the rules, which helps to understand the growth of the pattern. Another application is the Floradig software [5], which can record 3D coordinates and then convert them into geometric properties such as the internode length, and the angle between the main stem and branches. It also uses linear and non linear regressions to fit the data and provides geometrical attributes for the plant model.

However, we believe that what is missing from traditional L-System modeling approach is the sense of what is really going on in nature. In stead of doing trials and errors in adjusting L-System parameters and inventing grammars until the resulting 3D plant object look like the actual plant, can this actual parameters taken from the plant and integrated into a simple grammar? In doing so, the final output object should represent not only the plant but also how much we learn from the nature in creating the plant. Furthermore, the tool should allow real-time user interaction to help visualize the 3D object.

The proposed FractaL-Tree system consists of four modules as depicted in Fig.1. Starting with L-system generator, the axiom and rules were conducted to new string as defined by L-systems concept. Next, the coordinates and directions of model will be generated by Turtle interpreter module follow by decorating the rough model by

Geometry Generator and 3D renderer module. The decorated model is made interactive by 3D Renderer. Optionally, the standard format for the other 3D tools is made possible by the Wave-front OBJ Generator.

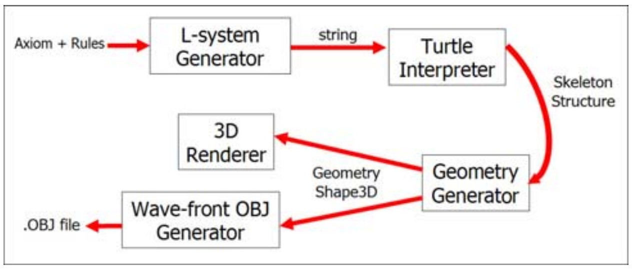


Fig.1 Components of the FractaL-Tree system.

2. Lindenmayer Systems

L-system is the grammar that used to generate new string by rewriting old string and following the syntax of production rules.

G	= {V, ω, P}
V	= Alphabet of system.
$\omega \in V+$	= is a nonempty word call **axiom**.
P $\in V \times V$	= is a finite set of **productions**.

A production (a ,X) $\in$ P is written as a -> X
The letter a and X are called the **predecessor** and the **successor** of production.

The following example shows the simulation of the development of a fragment of a multicellular filament found in the blue-green bacteria, Anabaena catenula and various algae. The filaments appear as a sequence of cylinders with a-type cells longer than b-type cells. To be manipulated by L-systems, the variables *a* and *b* are used to 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 following rules describe the development process of the filament:

$\omega : ar$
$p1 : ar \rightarrow albr$
$p2 : al \rightarrow blar$
$p3 : br \rightarrow ar$
$p4 : bl \rightarrow al$

Starting from a single cell *ar* (axiom), the following sequence of words is generated:

ar
albr
blarar
alalbralbr
blarblararblarar

The strings represent the structure of organism at each iteration as shown in Fig.2

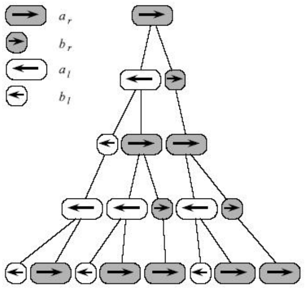


Fig.2 Development of a filament (*Anabaena catenula*) simulated by L-system.

3. 3D Turtle interpretation

The output string from L-system will be interpreted into 3D object structure by mapping each alphabet in the string into direction and operation of a turtle traveling in 3D space as displayed in Fig.3.

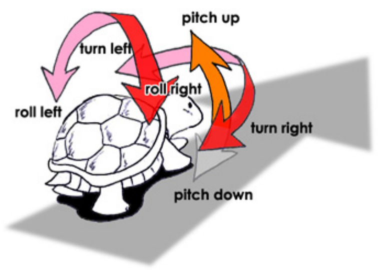
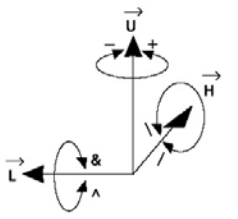


Fig.3 Turtle traveling in 3D space

The following operations control the turtle orientation in 3D space:

+	Turn left by angle α, using rotation matrix **RU**(α).
−	Turn right by angle α, using rotation matrix **RU**(−α).
&	Pitch down by angle α, using rotation matrix **RL**(α).
^	Pitch up by angle α, using rotation matrix **RL**(−α).
\	Roll left by angle α, using rotation matrix **RH**(α).
/	Roll right by angle α, using rotation matrix **RH**(−α).
|	Turn around, using rotation matrix **RU**(180◦).

The orientation of the turtle in space can be represented by three vectors H, L, U, indicating the turtle's heading, the direction to the left (or right), and the direction up (or down). Where R is a 3×3 rotation matrix. Specifically, rotations by angle *α* about vectors U, L and H are represented by the matrices in Fig.4.



$$\left[\vec{H'}\ \vec{L'}\ \vec{U'}\right] = \left[\vec{H}\ \vec{L}\ \vec{U}\right]\mathbf{R},$$

$$\mathbf{R_U}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R_L}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

$$\mathbf{R_H}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

Fig.4 Rotation matrixes.

4. Plant parameter description

An actual plant usually consists of many parts such as branches, trunks, leaves and flowers. To simplify the modeling task without losing much of reality, we will focus only on branches and trunks, and ignore the rest. We start by observing the **self-similarity** characteristic of the plant, then take a couple of real measurements as described in Fig.5 and below.

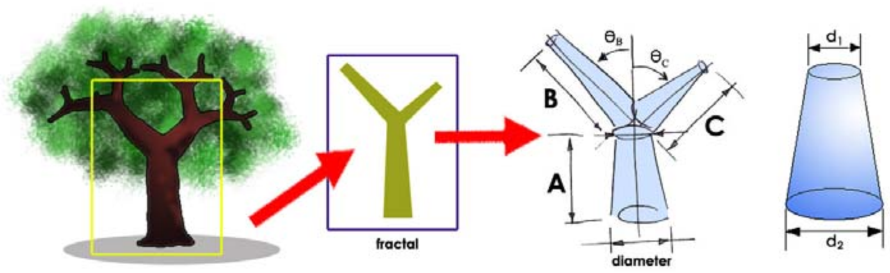


Fig.5 Measurement and plant characteristics extraction

We defined the grammar used with the example plant as follows:

Axiom:	F
Production rules:	F -> GG [-GF] [+F]
Phyllotactic angle:	45
Branch angle:	45

The parameters are defined as follows:
Length ratio: The proportion between the length of parent and a child branch.

From the example tree in Fig.5, the proportions between the branch length from the real tree are transferred into two fractions, B/A and C/A which are 1 and ½. The production rule is defined as F -> GG [-GF] [+F], corresponding to them while branch A, B, and C are represented by strings GG, -GF and +F. The number of alphabets in equivalent string are assigned to each branch length, length A =2; length B =2; and length C =1. By the examination, this production rule yields the proportion of the real tree.

Branching angle (Ɵ): The angle between parent branch and each child branch (Fig.5)

Phyllotactic angle (Φ): The angle between the plane of children branches (self-similarily pattern) and the plane of parent branch (Fig.6)

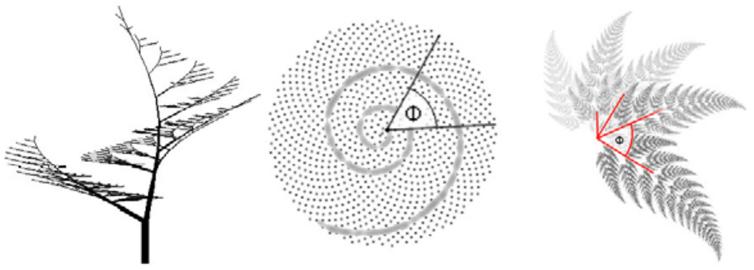


Fig.6 Phyllotactic angle

Diameter ratio: The parameter used to adjust the top diameter size according to the bottom diameter of each branch (d_1/d_2 in Fig.5)

5. Visualization development

The visualization application of development was implemented by **Java3D technology,** a standard extension to the Java 2 JDK [6]. This API provides a collection of high-level constructs for creation and manipulating 3D geometry and structures for rendering that geometry. It was designed for non expertise developer using packages that was provided for creation geometry model, visualizations, animations and interactive 3D objects. Other advantages are Java is convenient language for creation GUI applications and the performance of rendering 3D geometry is quite fast for display detail of model in real-time.

5.1 Geometry Generator

The coordinates, surfaces and meshes defining branches in space were added to the skeleton structure. The amount of the vertices controls the resolution of the plant model and increases the computational tasks. The shape of the plant is adjusted as following. The bottom diameter of the parent branch is proportion to the diameter and the number of the children branches. The top diameter of each branch is decreased from the bottom in proportion to the diameter ratio.

5.2 User Navigation & 3D Renderer

This module designed for user interaction, the plant model with different surfaces, polygon; hidden surface polygon; solid, and transparent could be viewed. Moreover, user can interact with model by provided functions in menu such as zoom in, zoom out, rotate, reset the best view, show or hide boundary box, capture the image, export model to Wave-front OBJ file format and adjust parameters for real-time simulation.

5.3 Wave-front OBJ Generator.

The practical options for using model in others 3D rendering software. We implemented the export engine for converting object to a standard 3D object text based file format that compatible with Wave-front OBJ file format [7]. This format supports both polygonal and Free-form geometry (curve and surfaces).

6. Experiment

The target plant with adequate self-similarity characteristic was chosen. Information obtained from the measurement is listed below (table 1) and the plant parameters were calculated according to these values.

Example: Alstonia scholaris tree.

Table 1 Plant parameters from a Alstonia scholaris tree used in this project.

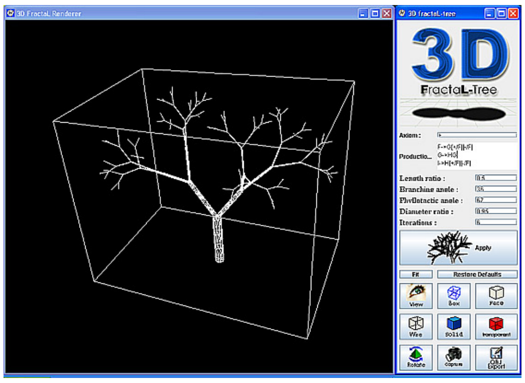
Parameters	Value
Length ratio	8 / 12
Branching angle	48,40,40,40
Phyllotactic angle	56
Diameter ratio	0.95

From this parameter values, we generated:

Axiom: F

Production rules: F->Y[++++++MF][-----NF][^^^^^OF][&&&&&PF]
M->Z-M
N->Z+N
O->Z&O
P->Z^P
Y->Z-ZY+
Z->ZZ

The grammar was put into the 3D FractaL-Tree program as illustrated in Fig.7 and result in plant structures proceeded by each iteration as shown in Fig.8.



Axiom : F

Productio... F->G[+/F][-/F]
G->HG
I->H[+/F][-/F]

Length ratio : 0.5

Branching angle : 36

Phyllotactic angle : 67

Diameter ratio : 0.95

Iterations : 6

Fig.7 A screenshot of the 3D FractaL-Tree program.

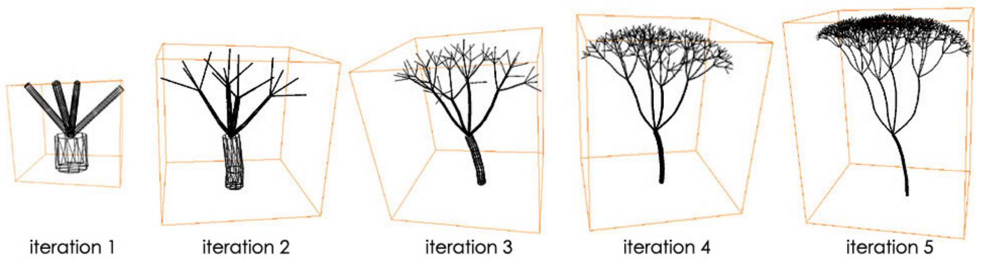


Fig.8 The structure of the example plant constructed by 3D FractaL-Tree from iteration 1 to 5

Fig.9 shows the real plant comparing to the constructed model. The model different from the real plant in the little amount of angle between the trunk and some branches.

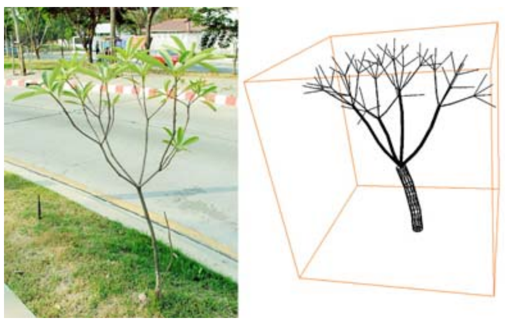


Fig.9 The real tree comparing to the constructed model.

7. Discussion

The traditional L-systems may result in plant model with high similarity to the nature. The systems tried to fit the model to target plant by adjusting parameters without the consideration of real measurement. Here, we proposed the new system taken into account the information retrieved from target plant which lead to more likeness and predictable model. However, the difference between the model constructed from our system and target plant remains. This is because of the complexity in nature which the random could not easily interpreted and calculated by human. Some obstacles may caused by the evolution for the plant to survive in stress conditions. The Stochastic L-systems may be used to construct model regarding the random in nature but high difference between the model and the target one may be received. Thus, the deterministic method is suitable to solve the growth pattern of plant. Our proposed system may in future be applied with various types of organism or DNA sequences to predict the folding of proteins. The application of L-System might be useful in the area of modeling growth development such as protein folding which is currently an active research field with extensive use of computing power. The L-System approach might provide a more economical solution

References

[1] Przemyslaw Prusinkiewicz, Aristid Lindenmayer, The Algorithmic Beauty of Plants., p.150 ,2004
[2] L-System Generation Program (LSystem4), http://www.trjjw.com/Lsystem4/L4Home.htm
[3] 3D L-Systems (Direct3D), http://www.alesdar.org/oldSite/IS/chap4-4.html
[4] LYNDYHOP, http://www.lab4web.com/chelmiger/lyndyhop/lh_start.html
[5] Floradig software, http://www.cpai.uq.edu.au/
[6] Dennis J Bouview, Java 3D API Tutorials, http://java.sun.com/products/java-media/3D/collateral/index.html
[7] Wave-front and Java3D .OBJ Format, http://www.javaview.de/guide/formats/Format_Obj.html