

Procedural fractal plants generation

Filip Rynkiewicz

Institute of Information
Technology, Lodz University of
Technology, ul. Wolczanska 215,
90-924 Lodz, Poland
173186@edu.p.lodz.pl

Piotr Napieralski

Institute of Information
Technology, Lodz University of
Technology, ul. Wolczanska 215,
90-924 Lodz, Poland

Abstract

Since the beginning of computer era we are trying to imitate nature. The results of those attempts are mathematical equations describing weather, snow flakes, plant growth or influence of species in individual biomes. In computer graphics fractal structure are often used because they can be simple characterize in mathematics. Those forms are common in nature. Self-similar fractals are created in computer graphics using Lindenmayer Systems. This article was formed to analyse efficiency of this algorithm in modelling 3D tree triangle meshes for games. The general motivation of this topic is to produce way to procedural modelling and simplified edition of complicated 3D tree models. By combining L-systems with Bézier curve there is a possibility to produce complicated 3D models. For example tree created with 468 branches having 87200 vertices and it is made in few seconds. In conclusion, time-consuming modelling in 3D programs can be replaced by solution described in this article.

1 Introduction

Using 3D modelling programs to create a complex 3D mesh of tree is very time consuming, because every branch has to be shaped separately. To speed up and automatize this process Lindenmayer System can be used. Since 1968, when Aristid Lindenmayer created grammar to represent simple multicellular organism, technique has evolved. Now L-Systems are used to manage 3D models of plants[1, 3, 5, 14, 4], fire visualization[2], music[6] or even neural web[7]. Variety of usage, flexibility and simplicity of this technique creates a possibility to extend it. Developers over years have used this technique to create software like L-studio[8], plugins to 3DsMax to animate plant growth[9] or in Houdini[11].

Most of the algorithms to create 3D tree mesh are generating static models, to change or move branch user must change parameters inside L-System, and recalculate whole model. To prevent unnecessary actions and calculation the *Bézier curve* was added as the tree skeleton. Adding possibility of changing Bézier points along Bézier curve, user can change every branch only by moving it's handler and recalculating only this alternated branch.

After long years of plants observations the conclusion was brought, every plant can be treated as a very complex fractal. The most important feature of this structure is self-similarity. In mathematics, a self-similar object is exactly or approximately similar to a part of itself (i.e. the whole has the same shape as one or more of the parts)[10]. Because of complexity of plants as fractals the simplifications must be applied. To do this the L-System can be used. Mentioned above technique is based on rewriting rules, so on replacing sub-terms of a formula with other terms. All formulas and terms are represented as strings, and L-system is all about changing one string to another based on rules. Every system has to have starting word, called axiom, rules of productions and number of iterations, so how many times rules must be applied to word.

2 Procedural Systems

Creating data algorithmically, based on mathematical equations, is the core of procedural system. This mean that program creates content on its own, or with limited or indirect user input. In case of L-systems there are few possibilities of creating procedural content using simple or more complicated L-systems.

2.1 OL-system

Most basic L-System is context-free L-system, called OL-system, where there can be multiple replacement rules to any given value. In example:

$$\begin{aligned} A &\longrightarrow \alpha \\ A &\longrightarrow \beta \end{aligned} \tag{1}$$

That means that A can be replaced with either α or β .

When there is one and only one replacement for the same letter in word, the system is called *DOL-system*, deterministic OL-system. Giving *axiom*(starting word) as b , rules : $b \rightarrow a$ and $a \rightarrow ab$ with 5 iterations simple DOL-system can be created. Example production of this system is shown at Figure1.

Because L-systems was inspired by nature, usage of it can be found in there. In example development of cyanobacteria *Anabaena Catenula* can be simulated by DOL-system.



Figure 1. Example of DOL system[12]

2.2 Turtle interpretation of string

L-systems are operating on symbols, so it's mean that it will always creates list of characters. Every symbol in created word can be classified as *steering character*, in example in Table1.

Table 1. Steering characters

sign	discription
+	Turn left by angle δ .
-	Turn right by angle δ
&	Pitch down by angle δ
^	Pitch up by angle δ
\	Roll left by angle δ
/	Roll right by angle δ
	Turn around, by 180°
[Add branch
]	End branch
[A ... Z]	Draw line between points

To create graphical content from it those characters must be parsed to specific action on *turtle*. *Turtle graphics* is term where turtle have it's position, angle which describes the direction in which it is facing and any other parameters that user will use, but position and angle are mandatory. When changing turtle behaviour, so parsing specific character, the different graphical content can be applied.

As it can be seen in the last column in Table1 the steering character to draw line between points can be represented by multiple upper cases letters. User can change behaviour of turtle by manipulating those characters. Can be defined which one are representing draw line and which are just to create randomness of given L-system.

2.3 Bracket L-System

OL-systems always create one long continued line. To add branching like behaviour, new characters must be added. Using $[$ and $]$ the *branches* can be created. First character is responsible for starting the branch and next for closing it. Branch is simply new line in geometric interpretation of string created by L-system Figure2.

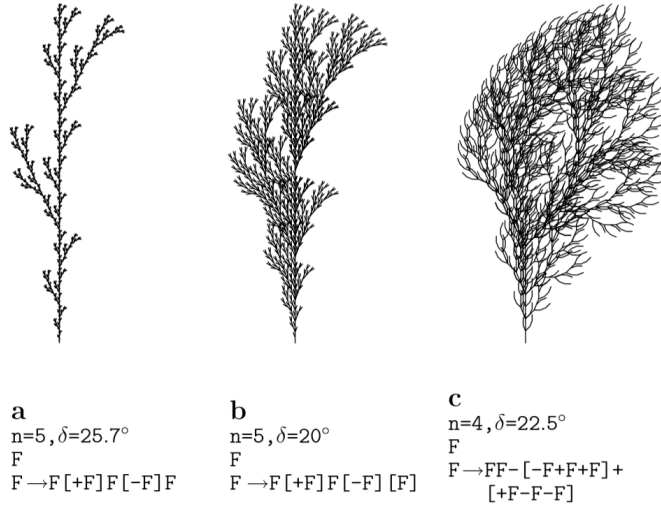


Figure 2. Example of Bracket L-system[12]

2.4 Parametric L-system

To create more complicated models the parametric L-system was developed. Thanks to parameters L-system can change the values of variables, in example the steering angle of branch. When in OL-system the sign $+$ was strictly assigned to hard-coded value, in this approach the value can be changed in every iteration using $+(r)$ where r is some kind of parameter. Every letter or string must have brackets immediately behind it and within them there can be multiple parameters, separated with comma, also mathematical expressions can be applied between parameters. Example:

$$C(x \cdot e, e \setminus 2, a + c, b - d, g) \quad (2)$$

As can be seen in Equation 2 the operands between parameters can be every mathematical operand that can be applied between two floating point numbers.

3 Bézier curves

Combining all mentioned above L-systems user can create model of plan, but without any possibility to change it without recalculating whole system. To add this feature in this work the Bézier curves was added. It will create whole skeleton of tree, and then based on it the mesh will be created.

Bézier curve is parametric curve, that can be generated under the control of other points. Approximate tangents by using control points are used to generate curve. The Bézier curve can be represented mathematically as

$$\sum_{k=0}^n P_i B_i^n(t) \quad (3)$$

Where P_i is the set of points and $B_i^n(t)$ represents the Bernstein polynomials which are given by

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i \quad (4)$$

Where n is the polynomial degree, i is the index, and t is the variable.

A B-spline curve is defined as a linear combination of control points P_i and B-spline basis function $N_{i,k}(t)$ given by

$$C(t) = \sum_{i=0}^n P_i N_i^k(t) \quad n \geq k-1, \quad t \in [tk-1, tn+1] \quad (5)$$

where

-

$$P_i = t_0, t_1, \dots, n, \quad (6)$$

P_i describes the control points.

- k is the order of the polynomial segments of the B-spline curve. Order k means that the curve is made up of piecewise polynomial segments of degree $k-1$.
- The $N_{i,k}(t)$ are the “normalized B-spline blending functions”. They are described by the order k and by a non-decreasing sequence of real numbers normally called the “knot sequence”.

$$t_i : i = 0, \dots, n+K \quad (7)$$

$N_{i,k}$ functions are described as

$$N_i^1(t) \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \text{ and } t_i < t_{i+1} \\ 0 & \text{Otherwise} \end{cases} \quad (8)$$

if $k > 1$

$$N_i^k(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1}^{k-1}(t) \quad (9)$$

also

$$t \in [t_{k-1}, t_{n+1}) \quad (10)$$

References

- [1] Prusinkiewicz P., Lindenmayer A., Hanan J., *Development models of herbaceous plants for computer imagery purposes*, SIGGRAPH Comput. Graph., New York, 1998
- [2] Zaniewski T., Bangay S., *Simulation and visualization of fire using extended lindenmayer systems*, AFRIGRAPH '03, Cape Town, 2003
- [3] Qi H., Qiu R., Jia J., *L-system based interactive and lightweight web3D tree modeling*, VRCAI '11, Hong Kong, 2011
- [4] Zhuming L., and King S., *Simulating tree growth based on internal and environmental factors*, GRAPHITE '05, Dunedin, 2005
- [5] Hanan J., Renton M., Yorston E., *Simulating and visualising spray deposition on plant canopies*, GRAPHITE '03, Melbourne, 2003
- [6] Manousakis S., *Musical L-Systems*, The Royal Conservatory, Hague, 2006
- [7] Palmer M., *Evolved neurogenesis and synaptogenesis for robotic control: the L-brain model*, GECCO '11, Dublin, 2011
- [8] L-studio, http://algorithmicbotany.org/virtual_laboratory/
- [9] Bartniak A., *3D animation of plant development using L-systems*, M.Sc. thesis, Lodz, 2013
- [10] Mandelbrot, B., *How long is the coast of Britain? Statistical self-similarity and fractional dimension*, Science, vol. 156, no. 3775, 1967, pp. 636–638. <http://www.jstor.org/stable/1721427>.
- [11] Houdini Engine L-System, <https://www.sidefx.com/docs/houdini/nodes/sop/lssystem>
- [12] Prusinkiewicz P., Lindenmayer A., *The Algorithmic Beauty of Plants*, Springer, 2004
- [13] Bézier curve, <http://mathworld.wolfram.com/B-Spline.html>
- [14] Fuhrer M. Hairs, *Textures, and Shades: Improving the Realism of Plant Models Generated with L-systems*, M.Sc. thesis, University of Calgary, 2005
- [15] Rozenberg G., Salomaa A. *The mathematical theory of L systems*, Academic Press, New York, 1980