Contribution title

Filip Rynkiewicz
Lodz University of Technology
Łódź, Poland
173186@edu.p.lodz.pl

Stanisław Nowak
MacGilll University
Montreal, Canada

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 Lindermayer Systems. This article was formed to analyze efficiency of this algorithm in modeling 3D tree triangle meshes for games. The general motivation of this topic is to produce way to procedural modeling and simplified edition of complicated 3D tree models. By combining L-systems with Bezier 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 modeling in 3D programs can be replaced by solution described in this article.

1 Introduction

Using 3D modeling programs to create a complex 3D model 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 grammatic to represent simple multicellular organism, technique has evolved. Now L-Systems are used to manage 3D models of plants[1][3][5], fire visualization[2], music[6] or even neural web[7]. Variaty os 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 the recalculate

the whole model. To prevent unnecessary actions and calculation the *Bezier Curve* was added as the tree skeleton. Adding possibility of changing Bezier Points along Bezier Curve, user can change every branch only by moving it's handler and re-calculating only this alternated branch.

After long years of plants observations the conclusion was brought, every plant can be treated as 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 subterms 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 axiom, starting word, rules of productions and number of iterations, so how many times rules must be applied to word.

2 Procedural Systems

2.1 DOL System

Deterministic and contex free L-system is one of the L-system, where context of the letter have no affect on algorithm. When axiom, starting word, is b and rules are: $b \rightarrow a$ and $a \rightarrow ab$ with 5 iterations, the example creation is shown at Figure1.

Figure 1. Example of DOL system[12]

2.2 Bracket L-System

Using additional character in words user can create extra behaviors. L-system always create one long continued line. 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.

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]

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

2.3 Parametric L-System

To create more complicated models the Parametric L-System was created. Thank to parameters L-system can change the values of variables, in example the steering angle of branch. When in DOL-System the sing + was strictly assigned to hard-coded value in this approach the value can be changed in every iteration using *+(r)* where r is some parameter. Every letter or string must have brackets immediately behind it and within them there can be multiple parameters from *0* ... *n* separated with comma. Every parameter can be a mathematical expression, parameters can be divided, multiplied, added, subtracted and etc. Example $C(x \cdot e, e \backslash 2, a+c, b-d)$

2.4 Turtle interpretation of string

L-System algorithm generating string. This string must be evaluated to create graphical content. Created word contains steering, drawing and omitted characters. The steering characters are the one which controls the behavior of turtle. Drawn are those which simply adding a line between points. *Turtle graphics* is term where turtle have it's position and angle which describes the direction in which it is facing. Sterring characters :

+ 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.

3 Methods

4 Results

Example 1

Rules:

$$F(l) \rightarrow F(l \cdot 2)$$

$$X(l) \rightarrow F(l)[/(r)X(l)]F(l)[\backslash(r)X(l) - (r)X(l)]F(l)[\backslash(r)X(l) + (r)X(l)]$$

Axiom:

$$\mathrm{X(1)}$$

Number of iterations:

$$5$$

Variables:

$$r \rightarrow 25$$

Example 2

Rules:

$$F(l) \rightarrow F(l \cdot 2)$$

$$X(l,w) \rightarrow F(w \cdot l)[/(r \cdot l)X(l,w) - (r)X(w,w)] + (r)F(l)[\backslash(l)X(l,w) + (r)X(l,w)] - (r)F(w)$$

Axiom:

$$\mathrm{X(1)}$$

Number of iterations:

$$5$$

Variables:

$$r \to 17.2$$

Example 3

Rules:

$$F(l) \to F(l \cdot 2)$$

$$X(l,w) \to F(w \cdot l)[/(r \cdot l)X(l,w) - (r)X(w,w)C(l)] + (r)F(l)[G(l) \setminus (l)X(l,w) + (r)X(l,w)] - (r)F(w) - (r)F(w)$$

$$G(l) \to F(l/5)[X(l,l) + (r \cdot k)]$$

$$C(l) \to F(l/5)[X(l,l)/(r \cdot k)]$$

Axiom:

$$\text{X(1,2)}$$

Number of iterations:

$$5$$

Variables:

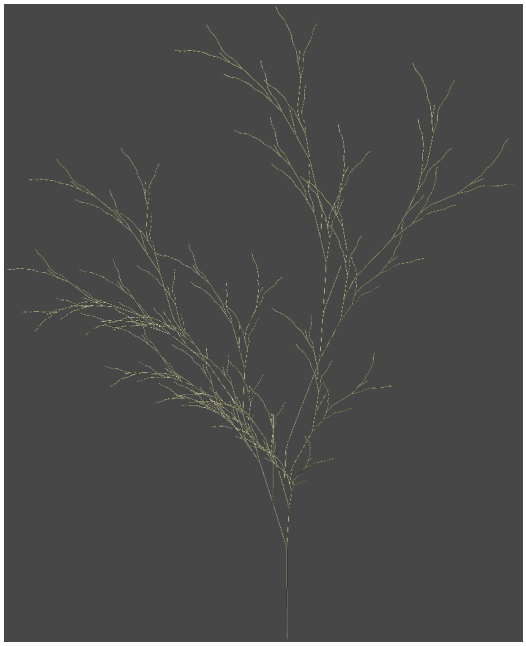
$$r \to 23.5$$

$$k \to 0.707$$

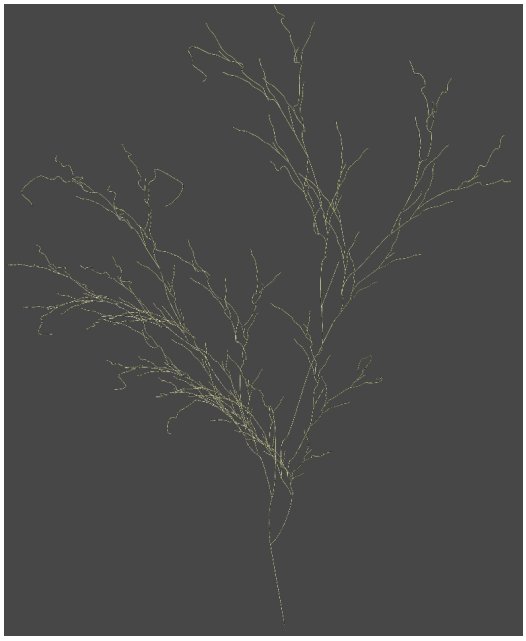
(a) Mesh without modification.
Own source.

(b) Mesh after modification.
Own source.

Figure 3. Comparison of meshes for example 1.



(a) Mesh withouth modification.
Own source.



(b) Mesh after modification.
Own source.

Figure 4. Comparison of meshes for example 2.

(a) Mesh without modification.
Own source.

(b) Mesh after modification.
Own source.

Figure 5. Comparison of meshes for example 3.

5 Discussion

For each of the example the same values was used. Five segments on the side surface and five points on spline, it's mean that every curve was divided on five equal parts. Each result of implemented algorithm was divided into two pictures. Screenshot of model generated by algorithm and model after changes. Changes was made by transforming Bezier Points on corresponded to branch Bezier Curve. First all the example and it's description was shown, then the pictures.

In first example for 5 iterations the number of branches is 468, word forming tree have 46086 chars and number of vertices in model is 87200.

This example is modification of simple fractal plant described in [12]. Every branch is directed at 25 ° relative to parent branch. As it can be seen at the Figure3a, the model created by algorithm is complex, using only two rule and one variable. Model after changes have been shown at Figure3b. Every branch in tree have been altered according to user actions.

Next example have 171 branches, world forming tree have 24918 chars and 73700 vertices in model. In this case the custom and longer rules have been used to make model looks like non-deterministic and more chaotic. Model before changes was shown at Figure4a, and after changes at Figure4b. In first example every variable had exactly one parameter without any mathematical operations, except $F(l \cdot 2)$. But in the second example every operation on parameter had to be evaluated and changed to float value, so the time of the calculation had increased significaly. Thanks to usage of multiple parameters and operation between them user can create more complicated structures, using less rules, but for the price of time for parameter evaluations.

Last example have 214 branches, world world forming tree have 24918 chars and 73700 vertices in model. POOOOOOOPRAW!!!!! Figure5a presents unchanged model of last example, Figure5b same model after changes. Creating more complicated rule, based on those in example 2, and adding two more simple rules the model in example 3 is more non-deterministic and tree-like than other examples. Time for creating this tree is not greater than time for example 2, because to store all rules algorith use C# *Dictionary* where the *Key* is from what chars rule must be changed, and *Value* stores to what rule chars must be changed.

6 Conclusion

By using technique described before the user can simply generate tree-like model in short time, small amount of vertices, and with ability to change every point of model using Bezier handles. Creating model with 300 branches using normal

3d modeling software would be very time consuming.We can move our work to machine and automatize process of modeling, using simple parameterized mathematical equations, without even knowing how the 3d modeling software works. And we can do this inside game engine, without other programs.

To evaluate all string into float values the simple parser have been created. Parser created with this implamentation is simple, but creating possibility to evaluate more than 3 parameters, and operations betwen them, using approaches from this implemantation would be very difficult. To ensure that every parameter would be evaluated propertly whole implementation will be rewriten into new *Boost Spirit X3* C++ library, and created as *Dynamic-Link Library*. Unity,Enreal Engine and others game engines have ability to write plugins based on *DLL* written in **C++**, so every engine could have their own implementation of this algorithm, based on *DLL*.

References

[1] Development models of herbaceous plants for computer imagery purposes,`http://dl.acm.org/citation.cfm?id=378503&CFID=841465770&CFTOKEN=79563252`

[2] Simulation and visualization of fire using extended lindenmayer systems

[3] L-system based interactive and lightweight web3D tree modeling

[4] Simulating tree growth based on internal and environmental factors

[5] Simulating and visualising spray deposition on plant canopies

[6] Musical L-Systems

[7] Evolved neurogenesis and synaptogenesis for robotic control: the L-brain model

[8] L-studio, `http://algorithmicbotany.org/virtual_laboratory/`

[9] Anna Bartniak

[10] How long is the coast of Britain?Statistical self-similarity and fractional dimension

[11] `https://www.sidefx.com/docs/houdini/nodes/sop/lsystem`

[12] Prusinkiewicz P., Lindermayer A. (2004) The Algorithmic Beauty of Plants. wersja elektroniczna.

[13] Fuhrer M. Hairs, Textures, and Shades: Improving the Realism of Plant Models Generated with L-systems. M.Sc. thesis, University of Calgary, August 2005. wersja elektroniczna

[14] Runions A., Lane B., Prusinkiewicz P. Modeling Trees with a Space Colonization Algorithm,Department of Computer Science, University of Calgary, Canada ,Eurographics Workshop on Natural Phenomena (2007).

[15] MacMurchy P., The Use of Subdivision Surfaces in the Modeling of Plants, THE UNIVERSITY OF CALGARY, April, 2004

[16] Smith C., On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling, THE UNIVERSITY OF CALGARY, April, 2006

[17] http://www.digitaltutors.com/tutorial/570-Procedural-Tree-Growth-Using-L-systems-in-Houdini

[18] `http://www.speedtree.com/images/modeler_lrg.jpg`

[19] `https://unity3d.com/5`

[20] `http://www.zobaczycmatematyke.krk.pl/025-Zolkos-Krakow/b-spline.html`

[21] Rozenberg G., Salomaa A. *The mathematical theory of L systems*, Academic Press, New York, 1980